Adapting Models of Visual Aesthetics for Personalized Content Creation

Antonios Liapis, Georgios N. Yannakakis *Member, IEEE*, Julian Togelius *Member, IEEE*

***Index Terms*—Computational Aesthetics, Experience-Driven Procedural Content Generation, Constrained Optimization, Interactive Evolution**

***Abstract*—This paper introduces a search-based approach to personalized content generation with respect to visual aesthetics. The approach is based on a two-step adaptation procedure where (1) the evaluation function that characterizes the content is adjusted to match the visual aesthetics of users and (2) the content itself is optimized based on the personalized evaluation function. To test the efficacy of the approach we design fitness functions based on universal properties of visual perception, inspired by psychological and neurobiological research. Using these visual properties we generate aesthetically pleasing 2D game spaceships via neuroevolutionary constrained optimization and evaluate the impact of the designed visual properties on the generated spaceships. The offline generated spaceships are used as the initial population of an interactive evolution experiment in which players are asked to choose spaceships according to their visual taste: the impact of the various visual properties is adjusted based on player preferences and new content is generated online based on the updated computational model of visual aesthetics of the player. Results are presented which show the potential of the approach in generating content which is based on subjective criteria of visual aesthetics.**

I. Introduction

In most design activities, the creator takes specific aesthetic preferences (her own or those of e.g. a customer) into account while creating their piece — be it a work of art, a household appliance or a component of a computer game. Their design choices may be constrained by the affordances of the medium (a newspaper cartoonist has a limited palette of colors), by the desired function (a table requires a flat surface) or by the perspective of cost-efficiency (for designs destined for mass-production). On the other hand, their preferences may be purely stylistic: the designer may want to elicit a specific emotion from their poem, may want to embody a specific art style into their painting, or may want to balance form and function while compromising neither (as with the case of computer gadget design). These aesthetic preferences guide the designer's pen or the artist's brush throughout the creative process; as soon as the creation is complete, it is evaluated by their peers, their customers and the general public. If the creator's preferences match those of their critics, then the creation is well received; if the creator, however, does not share the same preferences with the public, then the creation is shunned and discarded.

Evolutionary design intends to substitute (or at least assist) a human creator with a computer program which generates content: the type of content can be art, engineering schematics, furniture or computer game elements such as weapons [1], levels [2] or rules [3]. Just as a human creator, so must the computer program take into account affordances, constraints and stylistic preferences while creating such content. In such evolutionary design projects, programmers *de facto* insert their personal preferences into the evolutionary process; as in traditional art and design, the creator assumes that the inserted preferences are representative of the preferences of the public in general. While market research informs human creators of the public's wishes, a computer program evolving designs has the potential of assessing their users' preferences by interacting directly with them. To a certain extent, interactive evolution achieves that in the same way that an art critic determines if a finished work is good or not. However, it is much more useful to understand the reasoning behind the user's response: what drove them to like a specific design and discard another? By understanding the underlying causes of these choices, such a program could guide the creative process based on the user's preferences (as opposed to the creator's).

This paper introduces an approach towards realizing such a program. Using procedurally generated game content which is contingent on both its functionality and visual appeal — in this paper the content is 2D spaceship designs for a hypothetical space combat game — it aims to establish a range of personalized visual properties which guide the generative process. Using these quantifiable visual properties as the fitness function of a genetic algorithm, the content generator can optimize game elements with the desired visual patterns as dictated either by a designer (offline) or a user (online). This paper focuses on how, by observing the choices of a user, the algorithm can discern the underlying factors affecting their choice and focus on those visual properties for generating future content. Our general approach is unique as: a) we follow a two-step adaptation procedure for the generation of personalized content adjusting both the content generated but also the computational model that assesses content quality; and b) we combine neuroevolution with constraint satisfaction in order to create content which fulfills some minimum requirements while optimizing aesthetic [4] or functional [5] properties. The ongoing adjustment of the focus of the generative process is expected to eventually lead to personalized visual aesthetic computational models and furthermore to the creation of high-quality content matching the personal preferences of the user.

This paper builds upon previous work [4] and extends it with a new representation, a different range of visual properties and

Authors are with the Center for Computer Games Research, IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark. Emails: {anli, yannakakis, juto}@itu.dk

a more effective method for adapting the aesthetic model. By enforcing certain desired properties in all generated spaceships through the representation, many feasibility constraints are alleviated and optimization becomes more efficient. The new representation aids visual identification of the generated content as spaceships, allowing for a better visualization to players adapting their aesthetic model based on the selection of such content. Finally, the user experiment presented in this paper significantly differs from the early prototype [4], ensuring more interesting choices are presented to the users, a significantly faster generation of novel content and a more direct (and visually apparent) adjustment of the aesthetic model to the participant's selection. Complementary previous work [5] has focused on the optimization of a spaceship's performance in a space combat game, which can be combined with the optimization of a spaceship's visual properties presented in this paper for an even more inclusive measure of content quality.

The presentation of the paper is as follows: Section II places the proposed methodology in the context of ongoing and previous work on evolutionary design and visual perception, while Section III presents the components of the proposed constrained optimization algorithm. Section IV describes the domain-specific methodology followed for the representation of generated game content, the satisfaction of its constraints, the visual properties being evaluated and the process of adapting an aesthetic model to player choices. Section V presents results of offline (non-interactive) optimization of a sample of visual properties and an experiment in which different users adapted their aesthetic model during the online evolution of new content. Section VI discusses the insights gained from the study and provides directions for future work; the paper concludes with Section VII.

II. Related Work

This section places the proposed methodology for the generation of personalized computational models of visual aesthetics (and content which is driven by those models) in the literature. As this methodology builds upon the adaptation of game content to a specific player's experience, a short description of Experience-Driven Procedural Content Generation framework [6] is provided; moreover, the adaptation of content based on a player's selection places it in the domain of online evolution. Finally, we present studies on visual perception which inspired the visual properties chosen to be evaluated and optimized in this paper.

A. Experience-Driven Procedural Content Generation

The game industry has in many cases preferred procedurally generated to author-created content in order to increase the unexpectedness or unpredictability of a game (and therefore increase its replayability value) in games such as *Diablo* [7] (for dungeons), *Borderlands* [8] (for items) or *Civilization* [9] (for the world map). In recent years, the procedural generation of content is also used during the development of a game to limit development time and cost, with applications like *SpeedTree* [10] and *WorldMachine* [11].

Despite its long history within the game industry, the procedural generation of game content has only recently received attention from the academic community. Experience-Driven Procedural Content Generation (EDPCG) [6] is a novel approach to procedural content generation geared towards optimizing the experience of the player. EDPCG is synthesized by four main components: a Player Experience Modeling (PEM) component, a Content Quality component (which evaluates the generated content based on the PEM), a Content Representation and a Content Generator component which usually follows a search-based PCG method [12]. The principal novelty of EDPCG over traditional search-based approaches is the identification of the Player Experience Model, which can be derived from explicit player reports [13] (subjective PEM), from physiological signals [14] or other modalities of player response [15] (objective PEM) or from player actions within a game environment [1] (gameplay-based PEM).

B. Interactive Evolution

While many EDPCG projects use an ad-hoc designed fitness function to assess content quality, others use interaction with a human to guide evolution. Interactive Evolutionary Computation (IEC) is "the technology in which EC optimizes the target systems based on subjective human evaluation as fitness values for system outputs" [16] and is used extensively for content whose quality is subjective and difficult to quantify. At its core, IEC requires a human user to select individuals which will breed to create a new generation. IEC is limited by the fact that user interest drops as the number of choices they have to make increases. In order to avoid *user fatigue*, most IEC projects find shortcuts for reducing the number of choices imposed on their users.

Interactive evolution traditionally requires a user to select (or rate) one or more options among a range of presented content, with the selected individuals receiving preference for selection; this method has been predominantly used for generating graphics [17], [18], [19] or music [20] but also for game content such as buildings [21] and race tracks [22]. In the current literature, IEC is used within EDPCG either to provide an indirect player model based solely on gameplay metrics (side-stepping user fatigue) [1] or to model a direct mapping between the content and a desired player experience which is provided either explicitly (e.g. through self-reports) [23] or implicitly (e.g. through biofeedback) [24].

C. Universal principles of visual perception

Many EDPCG (and evolutionary art) projects argue that interactive evolution is a necessity, since purely stylistic or aesthetic preferences are very difficult to recognize. However, research in cognitive psychology and neurobiology has established certain universal properties of form which are ingrained in human perception and important factors of visual taste; such properties determine the visual impact of an object.

In his book *Art and Visual Perception* [25], cognitive psychologist Rudolf Arnheim observes the psychological impact of certain art pieces on the viewer by assuming a holistic perceptual processing of the scene. Introducing the term *perceptual forces* as the psychological and physical forces that guide the viewers' attention at specific points and along specific axes on an object or scene, he attempts to identify the most important contributors to the creation of these forces: the simplest of those are balance and shape. For Arnheim, the main contributors of *balance* are weight and direction: weight refers to the pull of the viewer's attention on specific areas and is influenced by location (with more importance given to the image's center and the horizontal and vertical axis), while direction guides the viewer's attention along specific axes. On the other hand, Arnheim approaches *shape* in the context of the minimal visual cues that can accomplish identification. He attributes the perception of shape to simplicity, subdivision, similarity and difference. Simplicity is achieved when structural features of the shape are arranged in an easily deductible and memorable pattern; such structural features "can be described by distance and angle" [25]. Subdivision refers to the human ability to group visual cues in order to dissect the whole into visually distinct parts. Similarity can visually group distinct shapes or features into a single unit or pattern, while difference is perceived as an anomaly and grabs the viewer's attention.

From a different scientific field, neuroscientists Ramachandran and Hirstein [26] has also suggested "speculative and arbitrary" laws of art; these eight universal laws, grounded mostly on empirical studies of the brain, are: peak shift, isolation, grouping, contrast, perceptual problem solving, symmetry, abhorrence of coincidence and metaphor. Of these properties of visual perception, which he identifies as common in all brains and thus resistant to cultural influences, this paper focuses on symmetry, peak shift and metaphor to identify the most influential features for spaceship identification.

D. Novelty of this paper

In the context of EDPCG research, the approach described in this paper introduces an inclusive player experience model (i.e. aesthetics computational model) which is dynamic as it is adjusted to the player's preferences during the interaction. Moreover, the approach provides an evaluation of visual quality rooted in theories of human perception, a versatile model for content representation which allows for a wide variety of generated shapes, as well as an efficient method for constrained optimization through the two-population paradigm described in Section III-B.

While the proposed approach of selecting the most preferable content among a range of presented content follows the paradigm of interactive evolution projects such as PicBreeder [17], unlike such projects it does not substitute a fitness function with a user. Instead of explicitly adjusting the fitness score of a selected individual, the user can — with a single selection — affect the way the fitness function is computed and therefore the fitness score of all individuals. Figure 1 presents the innovation of the proposed approach within the search-based PCG [12] framework. The adaptive aesthetic model presented in this paper provides a direct mapping between content and visual taste, limits the need of user input and therefore user fatigue and can estimate expected player satisfaction in previously unseen content.

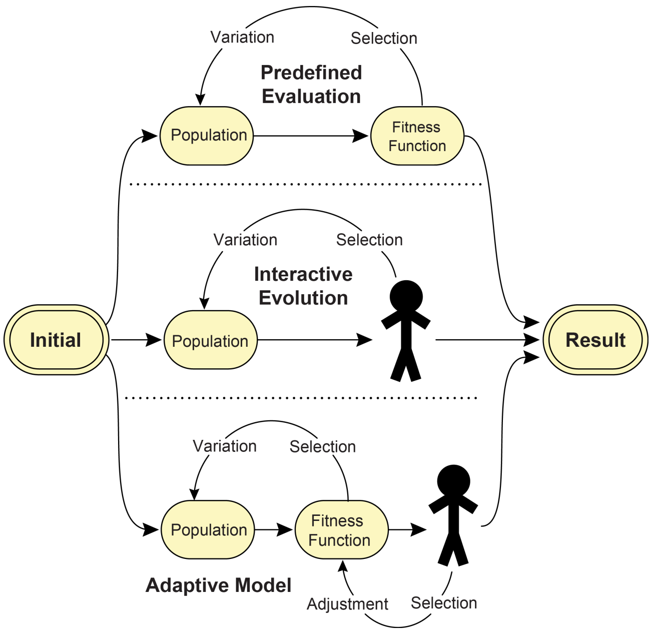


Fig. 1: The two dominant approaches to search-based procedural content generation (predefined-evaluation and interactive evolution), and the approach presented in this paper (adaptive model) which adjusts the fitness function based on user input; the fitness function in turn ranks content by a personalized measure of quality.

While the presented framework is inspired by the Galactic Arms Race game [1], it is distinct in that it evolves the spaceships themselves rather than their weapons, controls the generative process through constraints and proposes an indirect form of preference modelling. In the context of Smith's and Mateas' "meta-level design problem of sculpting an appropriate artifact design space" [27] which is tackled by automating the generation of the artifact design space through the definition of constraints and desired properties, the approach presented in this paper affords not only the satisfaction of constraints and optimization of one or more properties, but automates the interpretation of generated artifacts and the iterative refinement of the design space model.

Outside the domain of computer game content, many evolutionary art projects share this paper's motivation of a global or personalized aesthetic model.Although many such projects use interactive evolution for the selection of fit content [28], [17] or predefined evaluation functions which create artifacts of a specific style [29], [30], [31], a number of researchers have proposed methodologies with similarities to our presented approach. Jewelry Art Form Generator [32] implements a "designer interface" which allows the user to adjust their desirability of visual properties such as mirror symmetry or the golden ratio. While our approach uses similar descriptive terms for identifying and quantifying visual patterns, it allows the user to define the fitness function implicitly through the selection of preferred content rather than through a potentially unintuitive set of adjustable parameters. Baluja et al. [33] on the other hand use a neural network's output as a fitness score indicative of visual quality; the input of the artificial neural network (ANN) is the entire image and the ANN is trained on a set of computer-generated images. The failure of their approach to generalize (based on the large error in the test set) was an argument in our approach for the use of predetermined fitness functions for certain visual properties and adjusting their impact based on selected content. Machado and Cardoso [34] use an Artificial Art Critic (AAC) to evaluate 2D images from two complexity estimates; the AAC's evaluation is used as a fitness score for a content generator. Parameters in the AAC's evaluation formula can be adjusted directly by the user or the user "can indicate an image which he finds suitable and let the system set the optimimum values by estimating its complexity" [34]. The adjustment of the aesthetic model based on selected content has several similarities to our proposed approach; however, our approach uses a different representation (2D polygons' points rather than 2D images' pixels), different aesthetic properties and distinguishes between feasible and infeasible content. Most significantly, the adaptation process of our approach takes into account the user's unselected content and retains information from previous user choices, allowing for an iterative refinement of the user's visual taste.

III. Neuroevolutionary Constrained Optimization

This section presents the two main components of the neuroevolutionary constrained optimization algorithm used for the purposes of this study.

A. CPPN-NEAT

Introduced by Stanley [35], Compositional Pattern Producing Networks (CPPNs) are neural networks specifically designed to represent content with regularities, and which are capable of being optimized through artificial evolution. Assuming that development in nature consists of a series of progressively more localized *coordinate frames* (where a coordinate is a "conceptual device for describing an abstract configuration of any type" [35]), Stanley argues that development is analogous to a series of function compositions which transform the base coordinate frame to increasingly more localized coordinate frames with each transformation applied. This sequence of function compositions can be represented as a connected graph of such functions, with the initial coordinate frame as input and the most localized coordinate frames as output.

CPPNs can be optimized via neuroevolution of augmenting topologies (NEAT) [36]. NEAT starts evolution with a uniform population of CPPNs with the simplest topology (no hidden nodes) and random connection weights. As evolution progresses, more hidden nodes and links are added to the CPPNs; when a node is first added to the network, its activation function is selected randomly from a range of pattern producing functions (such as symmetrical or periodic functions). Genetic diversity is maintained through *speciation*, with individuals competing primarily with members of their own species, allowing them to optimize their structure without being overwhelmed by individuals of different species with more complex (and possibly more optimal) topologies.

B. FI-2Pop

While genetic algorithms have shown great promise in the domain of function optimization [37], the difficulties they face in solving constrained numerical optimization problems [38] has given rise to many different methods for handling such problems. The Feasible-Infeasible Two-Population (FI-2Pop) genetic algorithm [39] is a recent approach to constrained optimization through artificial evolution; its principle being the maintenance (throughout the execution of the algorithm) of two populations — one containing only feasible individuals and the other containing only infeasible. Each population selects and breeds only among its own members in order to optimize its fitness function, with each population having a different evaluation strategy. While the feasible population conducts its optimization in much the same way as in an unconstrained problem, the objective function of the infeasible population shifts the latter towards the boundary of feasible space, where the optimum solution often lies [38]. The proximity of infeasible individuals to the boundary of feasible space increases their chances of producing feasible offspring. The offspring of both generations are tested for constraint satisfaction, with infeasible offspring (regardless of whether their parents were feasible or infeasible) being inserted into the infeasible population and feasible offspring being inserted into the feasible population. This migration of offspring from one population to the other (an indirect form of inter-breeding) contributes to the variation of both populations; depending on the size of the feasible set, this migration may be the only source for feasible individuals.

The algorithm proposed in this paper evolves CPPNs through NEAT using both a feasible and an infeasible population, yielding a constrained optimization approach through neuroevolution.

IV. Methodology

For the purposes of this study, spaceship shapes (and indirectly their thruster and weapon topologies) are being evolved to satisfy several technical and design-specific constraints as well as to optimize their visual quality, which is determined based on various aesthetic principles. The model of visual quality used to evaluate generated content can be adjusted based on the spaceships chosen by a user. This allows the algorithm to adjust its focus to specific visual properties prevalent in one or more of the user's chosen shapes.

This section presents the process of the spaceship's generation and its evaluation. While this paper focuses on the evaluation of visual quality, generated spaceships are assumed to be inserted in a prototype 2D space shooter game consisting of planets (acting as obstacles) and enemy spaceships (see Fig. 2). Spaceships are expected to be able to function within such a game (move, shoot, avoid obstacles); "performance" in this section refers to these functionalities, and assumes a steering controller based on the spaceship's physical properties detailed in Section IV-A. For more information about the prototype 2D gameworld, the spaceship's intended functionalities, and the optimization process of the generated spaceships' performance, the reader is referred to earlier work by the authors [5].

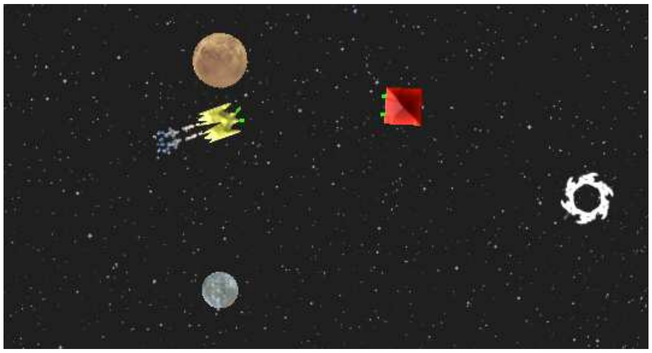


Fig. 2: A screenshot of the 2D space shooter in which spaceships will be used. Depicted are the procedurally-generated spaceship (yellow), an enemy spaceship (red), two planets acting as obstacles (the two spheres) and a goal area (white) towards which the test spaceship is moving.

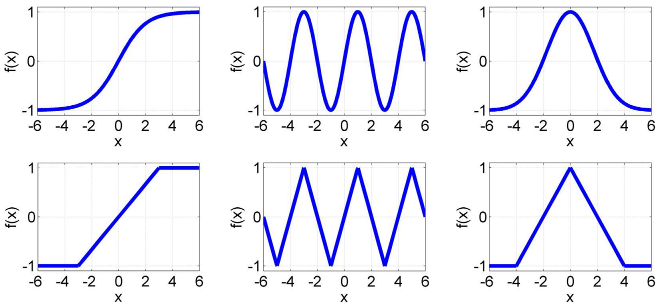
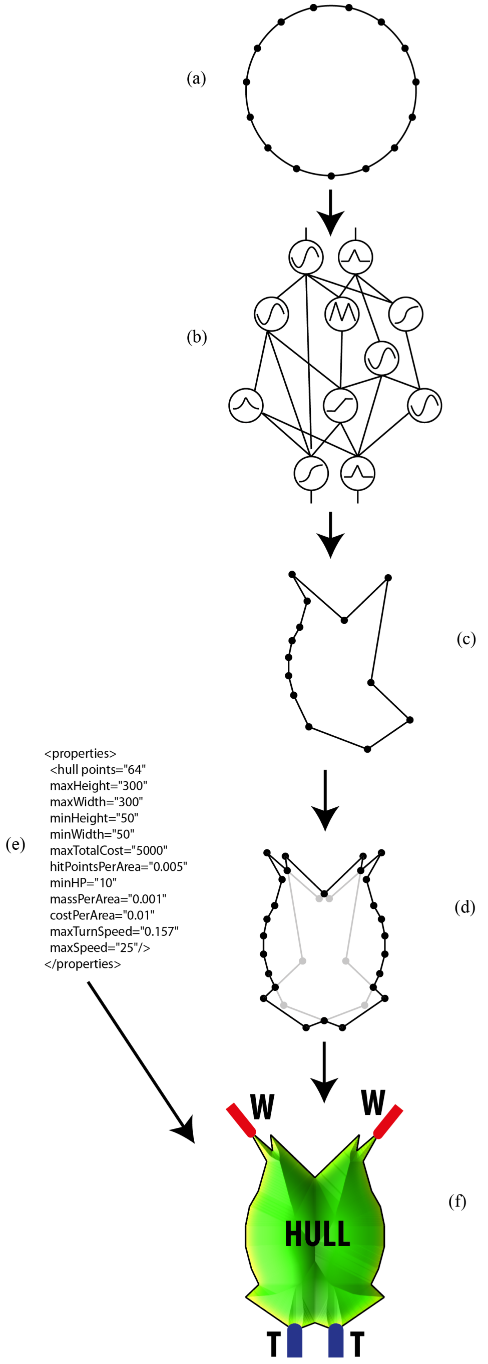


Fig. 3: Activation functions $f(x)$ used by the CPPN's nodes.

A. Representation

The generated spaceships are encoded as Compositional Pattern-Producing Networks (CPPNs) [35] which choose their nodes' activation functions among six options shown in Fig. 3. The CPPN receives a sequence of inputs in the form of 2D coordinates (corresponding to 15 equidistant points on a circle) and returns a sequence of 2D coordinates for the pattern of the spaceship's base shape. This "base shape" is subsequently mirrored along the vertical axis passing by its midpoint and joined with the original base shape, ensuring that the resulting spaceship will be symmetrical (see Fig. 4). The top-most points are assigned weapons aligned to that point's normal. The bottom-most points are assigned thrusters aligned to the vertical axis; any other thruster alignment would yield inferior spaceship performance. This representation differs from that presented in previous work [4], [5]: although it sacrifices representational freedom (with a single type of weapon and thruster), it increases the chances of feasible and well-performing individuals while the enforced symmetry increases the identification of generated results as spaceships. Given that the generated spaceships are expected to function (move, shoot) within a prototype 2D space shooter game [5], the enforced thrusters' symmetry simplifies the spaceship's physics model significantly, providing a much more sensible movement pattern.



```
<properties>
  <hull points="64"
  maxHeight="300"
  maxWidth="300"
  minHeight="50"
  minWidth="50"
  maxTotalCost="5000"
  hitPointsPerArea="0.005"
  minHP="10"
  massPerArea="0.001"
  costPerArea="0.01"
  maxTurnSpeed="0.157"
  maxSpeed="25"/>
</properties>
```

Fig. 4: Step-by-step generation of a spaceship. The input circle (a) is transformed by the CPPN (b) into a pattern (c) which is then merged with its reflection (d). The game parameters (e) determine the properties of the final spaceship (f). Weapons (W) and thrusters (T) are placed at the top and at the bottom of the spaceship, respectively.

The properties of attached weapons and thrusters (such as cost, thruster power, weapon cooldown or projectile damage) are stored in a collection of game-specific parameters, which also include constants such as mass per surface unit and maximum width and height of the spaceship (see (e) in Fig. 4). Once the spaceship has been generated according to the above procedure, its physical properties (such as the spaceship's

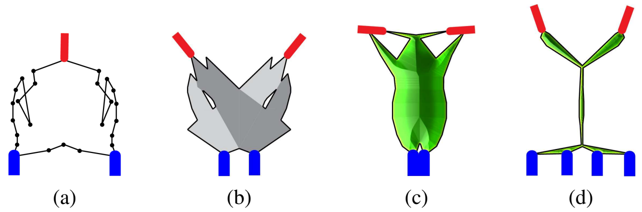


Fig. 5: Sample spaceships that fail different constraints: the spaceship in (a) has hull lines that intersect, in (b) it has holes (which are created during merging of the base shape, shown in light grey, with its reflection shown in dark grey), in (c) it has weapons which intersect with the hull and thrusters which intersect with each other. While the spaceship in (d) appears plausible, it exceeds the maximum limit imposed to the spaceship's speed (due to the small mass and large number of thrusters). Each constraint's distance from feasibility is measured in (a) from the number of hull line intersections, in (b) from the number of holes, in (c) from the number of intersections of weapons or thrusters and in (d) from the difference of the spaceship's speed and the upper speed limit.

mass, its acceleration and maximum speed) are determined based on its surface area and attached thrusters.

B. Constraint Satisfaction

Generated spaceships must fulfill a number of requirements in order to be able to function in a game. Such requirements may arise from the needs of rendering physics simulations (such as a non-degenerate polygon and positive mass), from the need of a spaceship appearing "plausible" to the user (such as weapons and thrusters that do not intersect with each other) and from the game design itself (which can impose a maximum number of weapons or a maximum speed). While not an exhaustive list, the most important constraints are shown in Fig. 5 along with sample spaceships which violate them. The inclusion of constraints elevates the problem of spaceship design to one of constrained optimization, which is handled by simultaneously evolving two populations of CPPNs. Using the Feasible-Infeasible 2-population paradigm [39], CPPNs encoding spaceships which satisfy all the constraints are inserted in the feasible population while CPPNs encoding spaceships which fail one or more constraints are inserted in the infeasible population. The feasible population uses NEAT [36] to optimize its members according to a measure of visual quality presented in Section IV-C, while the infeasible population uses NEAT to minimize its members' distance from a feasible solution. Each failed constraint has its own distance from feasibility which usually is a scalar value: Fig. 5 designates the measures used to calculate each of the sample constraints' distance from feasibility. The sum of every constraint's distance from feasibility constitutes the total distance from feasibility for a spaceship: if this value is 0 then the spaceship is feasible. Offspring of either population can be feasible or infeasible, allowing for a form of interbreeding which increases the diversity of both populations.

C. Visual Quality

Drawing inspiration from the works of Ramachandran and Arnheim documented in Section II-C, the presented framework is able to quantify a number of visual properties by parsing the polygons of the generated spaceships using the CGAL library [40]. According to the literature, some of the most important visual properties of the 2D spaceships are *symmetry*, *weight* as well as its outline. This paper identifies several significant visual properties of the spaceship's hull, although more aesthetic properties have been included in previous work [41]. Visual quality is assessed solely on the spaceship's hull (ignoring color, lighting and other aesthetic properties) since it has sufficient representational power to generate a large variety of shapes. Each of the mathematical formulas described below for quantifying the visual properties follows the format of $\mu(x)$, where x a value derived from parsing the spaceship's polygon and $\mu(x)$ a membership function which allows for granularity in the choice of optimal values for variable x.

Symmetry can be measured by reflecting the hull of the spaceship along an axis passing from its midpoint (see Fig. 6a). The fitness score for symmetry is computed as:

$$f_{sym} = \mu_{sym}\left(\frac{A_{\cap}}{A_{\cup}}\right) \tag{1}$$

where $A_{\cap}$ is the surface of the common area in the base and the reflected shape and $A_{\cup}$ is the surface of the area occupied by either the base or the reflected shape.

Weight (or weight distribution) can be measured by calculating the surface of a "focus" part of the spaceship's hull: example "focus" parts are displayed in Fig. 6b-6c. The fitness score for weight is computed as:

$$f_W = \mu_W\left(\frac{A_p}{A}\right) \tag{2}$$

where A_p is the surface of the "focus" part of the spaceship's hull and A is the surface of the entire spaceship's hull.

Containment builds on the notion of *weight*, but the "focus" part of the spaceship's hull is determined by a more complex shape acting as a "cookie cutter" (see Fig. 6e). The fitness score for containment is computed as:

$$f_C = \mu_C\left(\frac{A_c}{A}\right) \tag{3}$$

where A_c is the surface of the part of the spaceship's hull contained within the designated shape.

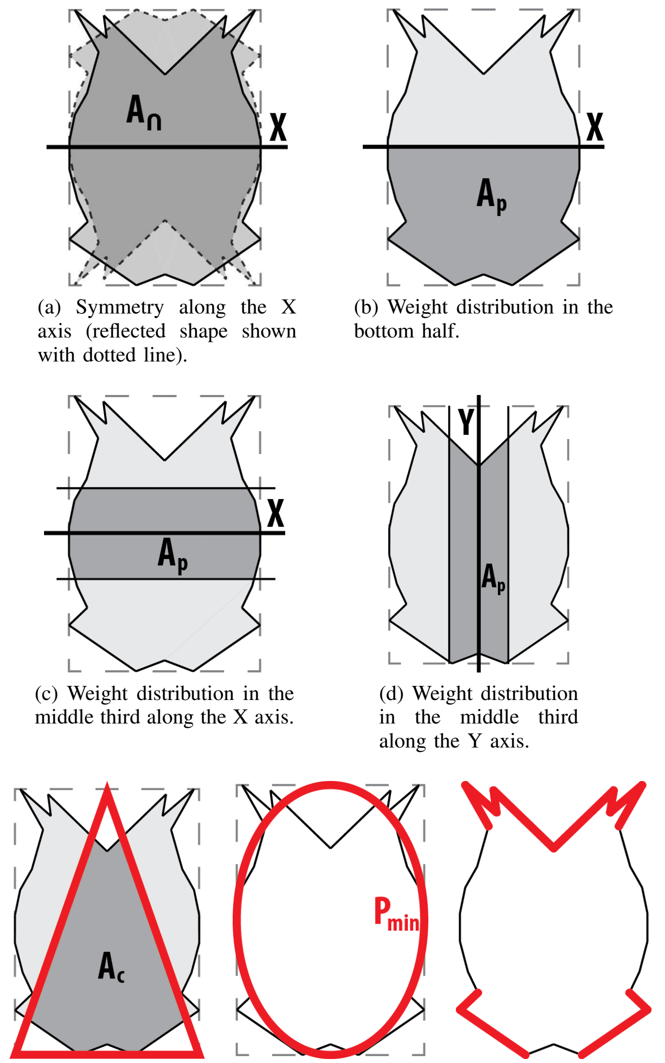
Simplicity rewards spaceships with simple outlines, whose hull's perimeter length is short. The fitness score for containment is computed as:

$$f_{sim} = \mu_{sim}\left(\frac{P - P_{min}}{2P_{min}}\right) \tag{4}$$

where P is the hull's perimeter length and P_{min} is the perimeter of an oval inscribed within the hull's bounding box (see Fig. 6f).

Jaggedness evaluates the presence of acute angles in the spaceship's outline. The fitness score for containment is computed as:

$$f_J = \mu_J\left(\frac{P_J}{P}\right) \tag{5}$$



(a) Symmetry along the X axis (reflected shape shown with dotted line).

(b) Weight distribution in the bottom half.

(c) Weight distribution in the middle third along the X axis.

(d) Weight distribution in the middle third along the Y axis.

(e) Containment within a forward-pointing triangle (red).

(f) Simplicity and the oval (red) within the bounding box.

(g) Jaggedness and the inscribed lines (red) forming sharp hull's angles.

Fig. 6: Visual properties: variable $A_\cap$ is the surface of the common area in the base and the reflected shape (see eq. (1)), A_p the surface of the "focus" part of the spaceship's hull (see eq. (2)), A_c the surface of the part of the spaceship's hull contained within the designated shape (see eq. (3)) and P_{min} the perimeter of an oval inscribed within the hull's bounding box (see eq. (4)).

where P_J is the length of all lines forming an angle between 20° and 60° or between 300° and 340° (lines sharing two such angles are not calculated twice). Figure 6g illustrates which lines' length contribute to P_J.

The individual fitness scores presented above can be used on their own to optimize a single visual property such as symmetry or simplicity, or can be aggregated into a weighted sum representing a more inclusive aesthetic model. By using a weighted sum as the feasible fitness, the constrained optimizer can create content with high scores in many different visual properties. The weighted sum of fitness scores ($\sum_i w_i f_i$ where f_i is the fitness score of a visual property i and w_i its corresponding weight) is identified as *aesthetic score* (F) and is normalized to $[0, 1]$.

D. Adaptive Model of Quality

Using a weighted sum for deriving a comprehensive measure of content quality allows for the weights of this quality approximation to be adjusted in a straightforward fashion based on in-game player choices. Through this weighted sum, the evolution's objective function largely subsititutes user input with a personalized aesthetic model and limits user fatigue.

Following the formula of interactive evolution projects with explicit user selection (such as PicBreeder [17]), the experiment presented in this paper is structured into a series of *iterations*. In each iteration a number of spaceships are presented to the user, who selects one of them as the visually preferable. The goal of the adaptive model is to reward visual properties with a high fitness score in the selected spaceship and low fitness scores in the unselected ones while penalizing visual properties with a low fitness score in the selected spaceship and high fitness scores in the unselected ones. Towards that end, the weight of a visual property i when the player selects spaceship S is updated by:

$$\Delta w_i = \alpha(f_{i_S} - \bar{f}_{i_U}) \tag{6}$$

where α is a weight update step (0.01 in the experiment presented in this paper), f_{i_S} is the selected spaceship's fitness score for visual property i and $\bar{f}_{i_U}$ is the average fitness score for visual property i among the unselected spaceships. Assuming we are adjusting the weight of the selected individual's fitness property i, equation 6 follows the key principles of the Widrow-Hoff [42] weight update rule.

As the interactive evolution experiment presented in this paper (see Section V-C) introduces the option of selecting no spaceship, eq. (6) is not directly applicable. In cases where the user selects no spaceship, the working assumption is that the aesthetic model used to rank and present content is completely off-track, but the user has no insight towards which visual patterns it should strive. The relaxation of the aesthetic model (by reducing the impact of weights on the final aesthetic score) will likely result in more "generic" spaceships, allowing the user a wider range of available visual properties to choose from. This relaxation is achieved by penalizing visual properties with a high fitness score in the presented spaceships if their weights are positive but rewarding them if their weights are negative. Towards that end, the weight of a visual property i when the player selects nothing is updated by:

$$\Delta w_i = -\alpha \bar{f}_i \frac{w_i}{|w_i|} \tag{7}$$

where α is the same weight update step as in eq. (6), $\bar{f}_i$ is the average fitness score for visual property i among all presented spaceships and w_i is the weight of the unadjusted aesthetic model.

The weights are adjusted until the selected spaceship has the highest aesthetic score F among those presented or when the aesthetic score difference between the highest scoring spaceship and the selected spaceship starts to increase. Once this adjustment is complete, an additional set of 100 weight updates is performed via eq. (6) or eq. (7): this process provides an additional fitness bias towards the selected spaceship, and ensures adjustments to the aesthetic model even if the user selected the highest scoring spaceship or no spaceship at all. Since the aesthetic score only measures the relative contribution of visual properties' fitness scores, the final adjusted weights are divided by $\sum_i |w_i|$ resulting in normalized weight values.

V. Results

This section presents the results of the neuroevolutionary constrained optimization algorithm when one or more sample visual properties are targeted. Experiments in Sections V-A and V-B do not allow for player interaction and use lengthy offline optimization runs using a predefined objective function. In these offline experiments the optimization behavior of CPPN-NEAT is compared with other neuroevolutionary approaches and subsequently it is further analyzed including sample optimized spaceships. While Section V-A demonstrates the types of spaceships favored by each of the individual visual properties, Section V-B introduces a few sample complex aesthetic models, illustrating how conflicting visual properties can hinder the algorithm's optimization progress. The studies conclude with a user experiment in online content generation using a player-dependent aesthetic model which gets adapted through human-computer interaction.

For experiments in this paper, seven different visual properties are selected based on the heuristics provided in Section IV-C:

f_1 evaluates symmetry along the horizontal (X) axis (see Fig. 6a), with μ_W in eq. (1) calculated from $\mu_{sym}(x) = x^n$.

f_2 evaluates weight in the bottom half (see Fig. 6b), with μ_W in eq. (2) calculated from $\mu_W(x) = min\{\frac{5x}{4}, 1\}^n$.

f_3 evaluates weight in the middle third along the X axis (see Fig. 6c), with μ_W in eq. (2) calculated from $\mu_W(x) = min\{\frac{5x}{4}, 1\}^n$.

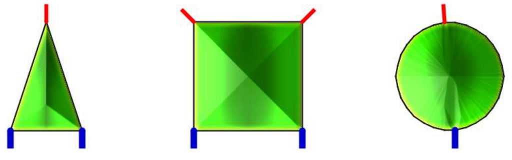
f_4 evaluates weight in the middle third along the Y axis (see Fig. 6d), with μ_W in eq. (2) calculated from $\mu_W(x) = min\{\frac{5x}{4}, 1\}^n$.

f_5 evaluates containment within a forward-pointing triangle (see Fig. 6e) with $\mu_C(x)$ in eq. (3) calculated from $\mu_C(x) = x^n$.

f_6 evaluates simplicity, with μ_{sim} in eq. (4) calculated from $\mu_{sim}(x) = min\{1, max\{x, 0\}\}^n$.

f_7 evaluates jaggedness, with μ_J in eq. (5) calculated from $\mu_J(x) = x$.

In the above heuristics of the seven visual properties, n is a parameter which puts pressure on highly fit content: $n = 3$ in all experiments presented in this paper. One exception is f_7, which does not include n in its membership function μ_J; preliminary tests illustrated that f_7 has very low scores for most spaceship shapes (as will be shown in following section) without the additional pressure from n.



(a) (b) (c)

Fig. 7: Ad-hoc baseline solutions used for comparative purposes against our approach: a triangle (a), a square (b) and a circle (c). Blue elements depict thrusters while red elements depict weapons.

TABLE I: Fitness scores of the three baseline solutions across all seven visual properties examined.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
Triangle	0.333	0.824	0.072	0.335	1.000	0.991	0.380
Circle	1.0	0.244	0.072	0.072	0.125	0.963	0.0
Square	1.0	0.244	0.140	0.147	0.167	0.993	0.0

In order to provide a set of baseline values for comparison with optimized spaceships' fitness scores, some obvious solutions to spaceship design are shown in Fig. 7 and their fitness scores for all seven visual properties are displayed in Table I.

A. *Offline Optimization of a Single Visual Property*

Each of the visual properties (f_1 to f_7) can be used on their own as a fitness function for feasible individuals in the constrained optimization algorithm. This section compares the optimization progress for a single visual property and shows that CPPN-NEAT affords a faster convergence over a number of neuroevolutionary techniques that also use the two-population approach. Subsequently it demonstrates the optimization process with CPPN-NEAT of the seven presented visual properties (f_1 to f_7).

1) Comparison among neuroevolutionary approaches: To evaluate the efficiency of our CPPN-NEAT approach, we compare it against a number of alternative neuroevolutionary mechanisms, all of which are using the FI-2Pop paradigm for constrained optimization and which, unlike CPPN-NEAT, employ only the hyperbolic tangent as their neurons' activation function. The alternative mechanisms are as follows:

- ANN-NEAT: this mechanism begins with minimal topology networks and augments their topology.
- EANN$_1$: this mechanism evolves the weights (without altering the topology) of a fully connected feed-forward network with no hidden nodes.
- EANN$_2$: this mechanism evolves the weights (without altering the topology) of a fully connected feed-forward network with two hidden layers of four nodes each.
- EANN$_3$: this mechanism evolves the weights (without altering the topology) of a fully connected feed-forward network with two hidden layers of ten nodes each. The choice for the EANN$_3$ topology came from an observation of the final evolved CPPNs, which had 19 hidden nodes on average.

The optimization progress of a single visual property (for space considerations, we are using f_2 as a sample fitness score)

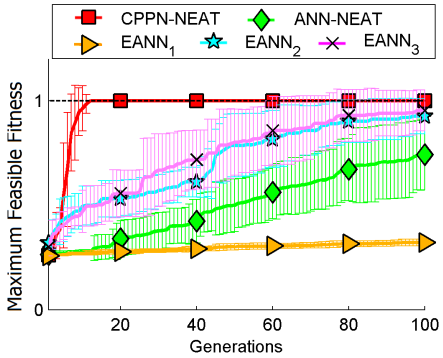


Fig. 8: Comparison of the optimization of the maximum fitness (using f_2) among different neuroevolutionary approaches.

for the different neuroevolutionary approaches is presented in Fig. 8. Similar results were obtained for all individual fitness scores examined. The results are gathered for the 5 different stochastic optimization mechanisms described above on a population of 250 individuals; statistics are calculated from 10 independent runs of the algorithms. The CPPN-NEAT approach obtains a significantly higher maximum fitness compared to all other approaches, except when compared with EANN$_3$ after 100 generations for which the p value equals 0.18. Significance is tested through standard t-tests (significance is 5% in this paper) at 10 generations, 50 generations and 100 generation as a representative sample of early, intermediate and prolonged optimization. Networks with EANN$_3$ topology are large enough to represent highly fit content, but the large number of parameters that their size entails makes their optimization slower compared to the augmenting topology approach of CPPN-NEAT. Of the other approaches, larger ANNs eventually discover fit content, ANN-NEAT suffers from unpredictable behavior (as indicated by the large standard deviation) while the minimal topology networks — due to their limited representational power — are unable to discover fit content even after 100 generations. Results obtained show the superiority of CPPN-NEAT for the problem examined as the mechanism demonstrates rapid and efficient design of highly fit content.

TABLE II: Fitness of the best individual at the beginning and the end of constrained optimization of a single visual property across 10 independent runs.

	First feasible		After 100 gen.		
	Mean	StDev	Mean	StDev	Max
f_1	0.9503	0.0243	0.9972	0.0043	1.0
f_2	0.2619	0.0098	1.0	0.0	1.0
f_3	0.2220	0.0887	1.0	0.0	1.0
f_4	0.1503	0.0654	1.0	0.0	1.0
f_5	0.2842	0.0515	0.9364	0.0313	0.9879
f_6	0.9844	0.0033	0.9954	0.0004	0.9958
f_7	0.0995	0.0528	0.9929	0.0139	1.0

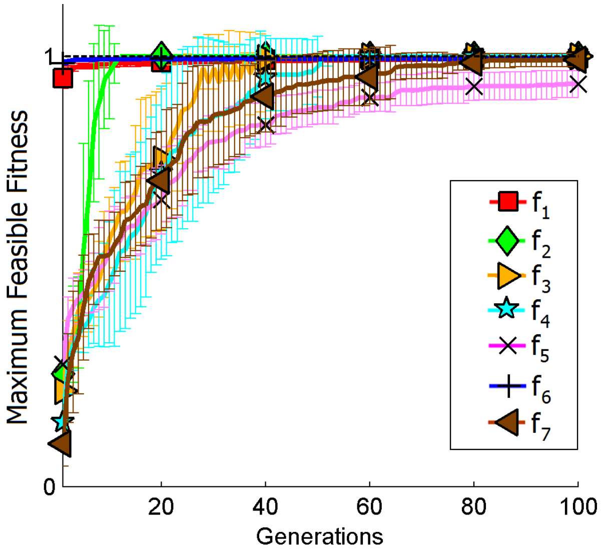


Fig. 9: Progress of the maximum fitness score in each population, during the optimization of a single aesthetic property (across 10 independent runs). The error bars designate the standard deviation among the different runs.

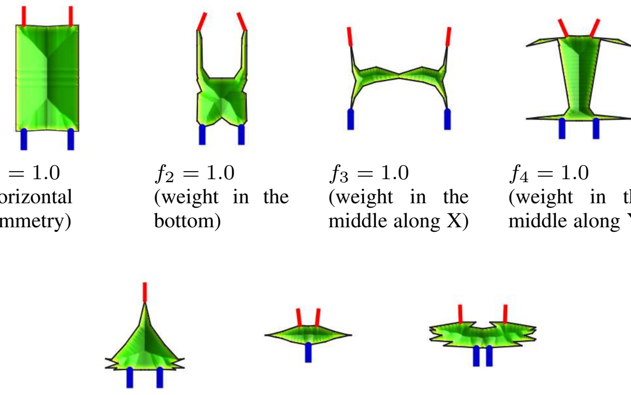
2) Generating content via CPPN-NEAT: Table II presents the fitness scores of the best feasible individuals at the beginning and the end of a constrained optimization process (after 100 generations), with a population of 250 individuals. The means and standard deviations are calculated from 10 independent runs. The first feasible individual in the population is used in the calculation of initial scores regardless of the generation it occurred. Fig. 10 presents the best final individuals among the 10 different runs, for each visual property. Fig. 9 illustrates the progress of the different fitnesses; combined with the information from Table II, it is clear that the visual properties of symmetry and simplicity (f_1 and f_6, respectively) have high scores even with simple networks. Because the points used as input to the CPPN are on a circle which has high scores in symmetry and simplicity (see Table I), the resulting spaceships — especially with simple networks which only apply subtle transformations to the initial coordinate frame — are more likely to be symmetrical (f_1) and very unlikely to have an "unbalanced" weight distribution (f_2, f_3 or f_4).

B. Offline Optimization of Multiple Visual Properties

While the optimization of a single visual property leads to highly fit content, it is only through the combination of different visual patterns that a meaningful spaceship shape can be identified. For space considerations, this section presents three sample combinations of the fitness scores f_1 to f_7 aggregated as a weighted sum with positive (1) or negative (-1) weights. The aesthetic combinations presented in this paper are identified as:

- $F_{A_1} = f_2 - f_6$
- $F_{A_2} = f_2 + f_4 + f_5 - f_6$
- $F_{A_3} = f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7$

The combinations of visual patterns were chosen in order to showcase the effects of genetic search in increasingly complex data: from the aggregation of two fitness scores in F_{A_1} to that of all fitness scores in F_{A_3}, the impact of searching the optima of many different (and possibly conflicting) visual patterns will become apparent below. The selection of aesthetic properties (and their weights) for F_{A_1} and F_{A_2} was made



$f_1 = 1.0$ (horizontal symmetry)

$f_2 = 1.0$ (weight in the bottom)

$f_3 = 1.0$ (weight in the middle along X)

$f_4 = 1.0$ (weight in the middle along Y)

$f_5 = 0.988$ (containment in a triangle)

$f_6 = 0.996$ (simplicity)

$f_7 = 1.0$ (jaggedness)

Fig. 10: Best individuals optimized for a single aesthetic property (across 10 independent runs).

after preliminary tests demonstrated that generated spaceships were more appealing when the simplicity aesthetic (f_6) was minimized.

1) Comparison among neuroevolutionary approaches: The efficiency of CPPN-NEAT is evaluated against four alternative neuroevolutionary methods using F_{A_1}, F_{A_2} and F_{A_3} as the objective functions; the optimization progress for these three aesthetic models is illustrated in Fig. 11. Results are collected for the five different neuroevolutionary mechanisms described in Section V-A, on a population of 250 individuals and the statistics are calculated from 10 independent runs.

For F_{A_1}, the CPPN-NEAT approach consistently finds the best possible individual within 100 generations; the CPPN-NEAT approach also has a significantly higher maximum fitness from all other approaches at 10, 50 and 100 generations, except when compared with EANN$_3$ after 10 generations for which $p = 0.06$. For F_{A_2}, the CPPN-NEAT approach achieves a significantly higher maximum fitness compared to the maximum fitness of all other approaches at 10, 50 and 100 generations, except when compared with EANN$_3$ after 10 generations for which $p = 0.35$. For F_{A_3}, all approaches have a very similar optimization progress throughout the course of evolution; only EANN$_1$ has a much slower progress since its simple topology fails to generate intricate patterns in its results. Excluding EANN$_1$, the differences between the maximum fitnesses for the other approaches are — save few exceptions — not statistically significant at 10, 50 and 100 generations. The key reason for this behavior appears to be the large number of (often conflicting) visual properties being combined; this problem will be discussed in the next section.

Observing the behavior of the different approaches in Fig. 11, the same conclusions as in Section V-A hold true. ANN-NEAT, however, shows an optimization behavior comparable to that of larger networks such as EANN$_2$ and EANN$_3$. It is unclear if the chosen aesthetic models suit the ANN-NEAT approach or whether the chosen preset networks are unable to represent highly fit content. Minimal topology networks (EANN$_1$) perform poorly, since their simple topology does not allow them to represent shapes with complex patterns. Overall, an important observation is the increasingly unpredictable behavior of neuroevolutionary approaches as the number of aggregated visual properties increases.

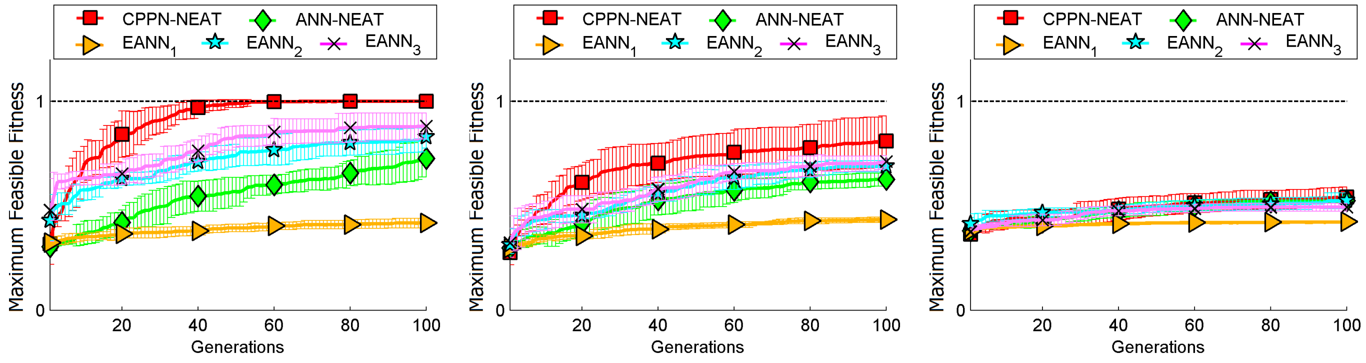
TABLE III: Fitness of the best individual at the beginning and the end of constrained optimization of multiple visual properties across 10 independent runs.

	First feasible		After 100 gen.		
	Mean	StDev	Mean	StDev	Max
F_{A_1}	0.3167	0.0973	1.0	0.0	1.0
F_{A_2}	0.2756	0.0595	0.8100	0.1184	0.9786
F_{A_3}	0.3629	0.0069	0.5403	0.0483	0.6131

2) Generating content via CPPN-NEAT: The results of the optimization process for the aesthetic scores F_{A_1}, F_{A_2} and F_{A_3} for 100 generations on a population of 250 individuals are shown in Table III collected from 10 individual runs. The progress of the best individual for the most successful run is presented graphically in Fig. 12a (for F_{A_1}), Fig. 12b (for F_{A_2}) and Fig. 12c (for F_{A_3}). Under each figure, the best individuals' phenotypes are shown on a 20 generation interval.

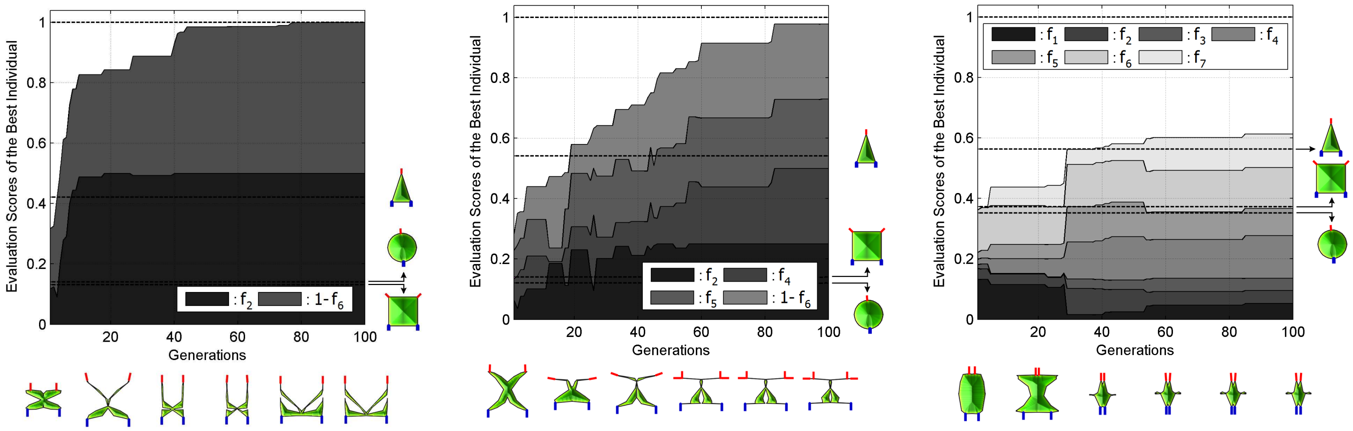
Results show an overall increase of the aesthetic score from its initial values; however, as the combined visual properties increase, so does the unpredictability of results. While optimizing a single property had a small standard deviation (see Table II), this deviation increases for two visual properties and moreso for four visual properties (see Table III). While Figures 12a and 12b show that all visual properties reach a high fitness score, in less successful runs certain visual properties dominated others. For F_{A_2} in particular, the final best individual often lacked a significantly positive score for the f_4 component.

For the sum of all visual properties (F_{A_3}), constrained neuroevolution fails to find a shape that maximizes the scores of all visual properties. Fig. 12c illustrates that even for the most successful run some visual properties (such as f_1, f_2 and f_3) have a low score while only f_4 has the maximum score in the final best individual; in less successful runs, some properties (primarily f_1 and f_3) have no score in the final best individual at all. It should be noted here that this behaviour is not only due to the failures of the aggregated approach at handling multi-objective evolution: the choice of visual properties being combined also affects the optimization progress. A shape cannot be at the same time symmetric along the X axis (f_1) as well as have its weight concentrated at the bottom (f_2) — the comparison between f_1 and f_2 of the square and the triangle in Table I illustrates the disparity. While a shape can in theory have its weight concentrated both in the middle third along the Y axis (f_3) and in the middle third along the X axis (f_4), such shapes are a very small subset of the optimal shapes either for f_3 or f_4, and therefore difficult to discover. Finally, a shape which could at least in part combine the intricate visual properties defined (not least of which is the requirement for having sharp edges as per f_7) would be impossible to have a simple outline (f_6). It is therefore, in part, the fault of the designer for requiring the simultaneous



(a) Optimization using F_{A_1} as the aesthetic model. (b) Optimization using F_{A_2} as the aesthetic model. (c) Optimization using F_{A_3} as the aesthetic model.

Fig. 11: Comparison of the optimization of the maximum fitness among different neuroevolutionary approaches.



(a) Optimization using F_{A_1} as the aesthetic model, which evaluates weight in the bottom (f_2) and *complexity* ($-f_6$ or negative simplicity).

(b) Optimization using F_{A_2} as the aesthetic model, which evaluates weight in the bottom (f_4) and in the middle third along Y (f_4), containment within triangle (f_5) and *complexity* ($-f_6$ or negative simplicity).

(c) Optimization using F_{A_3} as the aesthetic model, which sums all visual properties (f_1 to f_7).

Fig. 12: Stacked area plot of the best individual's progress for different aesthetic models of varying complexity. Under each figure, the best individuals' phenotypes are shown on a 20 generation interval. For legibility purposes, in plots 12a and 12b the progress of f_6 due to its negative weight is displayed as $1 - f_6$. The score of the baseline solutions are also included for comparison.

optimization of properties which are not compatible with each other. This could be alleviated by a more careful selection of visual properties and their weights (as was done for F_{A_2}), or by adapting these weights based on example shapes that epitomize the visual effect required. This is the rationale behind the adaptive model of visual aesthetics detailed in Section IV-D, an experiment of which will be discussed in the following section.

C. Online Adaptation of the Aesthetic Model

In order to evaluate the potential of interactive adaptation of the aesthetic model, a simple user experiment was conducted. This section presents the testbed used, the participants involved, the experimental process and a sample of the results gathered.

1) Setup and Participants: The online adaptation experiment took place in a controlled environment, on a single laptop computer: only one participant took the test at a time and, where possible, isolation of the participant from external distractions was preferred. A questionnaire given before the experiment included a brief description of the experiment's goal and a guide to running it.

In total 20 users participated in the test. Participants (15% female) were primarily selected from students of a game studies program. For that reason, the vast majority of participants play computer games, with 35% playing 20 hours or more weekly and 45% playing 5-10 hours weekly. Ages in the tested group ranged from 22 to 36 and nationalities included Greek (20%), Italian (15%), German (15%) and Danish (15%).

2) Method: Every participant was presented with the same eight spaceships (Fig. 13), loaded from a collection of space-

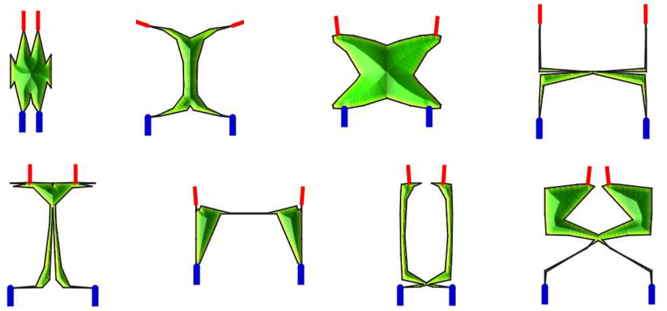


Fig. 13: Initial shapes presented to all participants, ordered from fittest (top left) to least fit (bottom right) as evaluated by the initial aesthetic model.

ships optimized offline for different visual properties and combinations thereof. This collection, consisting of 46 individuals (all of which are feasible), was the initial population from which new spaceships were evolved based on the player-adapted model of visual aesthetics. The aesthetic model was represented as a weighted sum of the seven visual properties (f_1 to f_7); the initial model's weight values were set to 1 for all participants. As presented in Section IV-D, the experiment is essentially a series of *iterations*: in each iteration a number of spaceships (up to eight) are chosen from the current feasible population and presented to the user. The presented individuals include the highest scoring spaceship and the lowest scoring spaceship, with remaining spaceships as evenly distributed as possible according to their aesthetic score. When there are not enough feasible individuals to provide a suitably diverse set, then the number of presented spaceships is smaller than eight. After the player selects a spaceship among the ones presented (or no spaceship at all), the weights of the aesthetic model are adjusted according to the process presented in Section IV-D. With the new aesthetic model, the current population evolves for 5 generations in order to optimize the re-weighted visual properties, and a new iteration begins with the presentation of a new set of spaceships among those evolved.

Figure 14 is a visualization of a participant's iteration, and illustrates the process of adapting the aesthetic model. The difference in the fitness scores of visual properties between the selected spaceship and the mean of the unselected ones heavily depends on the types of spaceships presented. In cases where the spaceships displayed have several similar visual properties, the player's selection greatly increases or decreases the weights of those visual properties that are different among the presented content. This allows the model to "focus" on one or two visual properties at a time, optimizing those on a population already optimized for previously prominent visual properties. The evolved content and the aesthetic model therefore complement each other; this may however cause stagnation if the population does not have very visually diverse results. The option to select nothing is designed to adjust the aesthetic model in a way that will allow diversity to be favored during the subsequent content evolution.

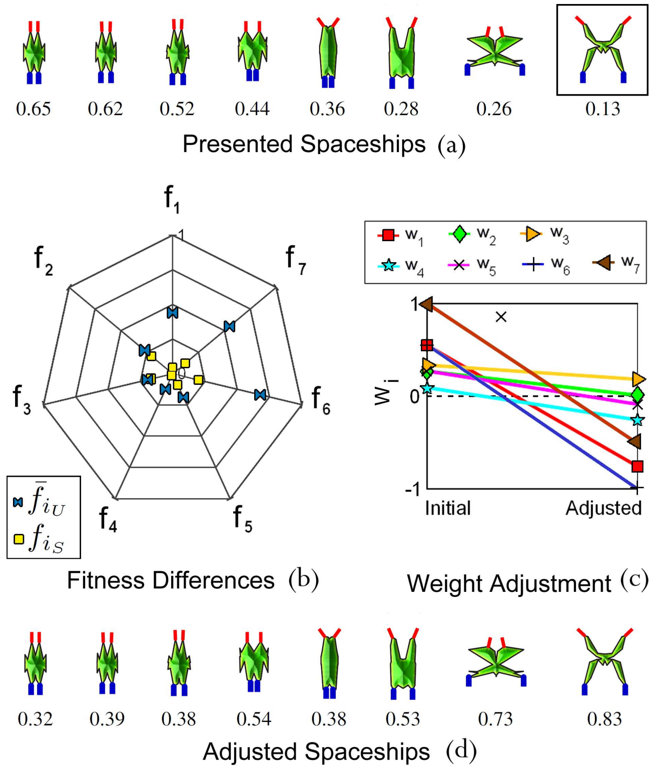


Fig. 14: An illustration of the adaptive modelling procedure: the weight update of participant No.5 for the 2^{nd} iteration. The presented spaceships and their initial fitness scores are displayed in (a) with the selected one inside a black outline. The difference between the selected spaceship and the mean of the unselected ones for each visual property is visible in (b). Based on this difference, the aesthetic model is adjusted until the selected spaceship has the highest fitness; the change in the weights is illustrated in (c). The fitness scores of the spaceships in (a), given the new aesthetic model, are shown in (d).

3) Results: The adaptation of the aesthetic model depends heavily both on the content presented and on the player's preferences; this makes a conclusive presentation of all participants' aesthetic models superfluous. Due to space constraints, Fig. 15 shows the aesthetic model's progress for the first 10 iterations of only one participant. The participant's selected spaceship as well as the aesthetic model's expected best spaceship are also illustrated over each iteration. The 1st iteration illustrates the initial aesthetic model provided to all participants (where all weight values equal 1). The shift from the 3rd to the 4th iteration showcases the effect of selecting no spaceship: the aesthetic model shifts most of its weights from positive in the 3rd iteration to negative in the 4th iteration, while the expected best spaceship in the 4th iteration is very different from that of the 3rd iteration. After the 5th iteration, the user's preferences become more consistent and more in tune with the expected best spaceships; this results in smoother changes in the aesthetic model, with the lack of symmetry ($-f_1$) being consistently the most significant contributor to the aesthetic score while simplicity (f_6) and jaggedness (f_7) alternate between positive and negative weights between iterations.

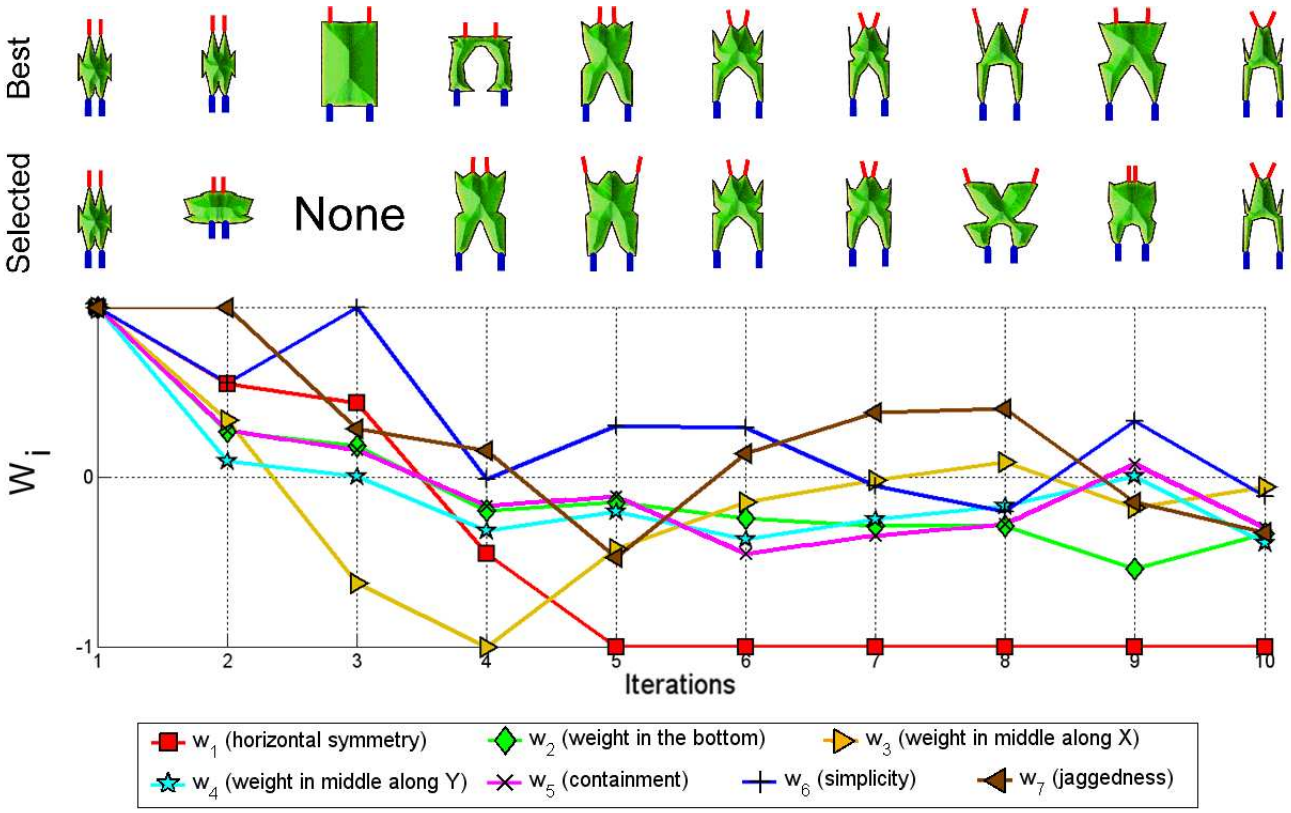


Fig. 15: The aesthetic model's progress over the iterations for a sample participant (participant No.9); the aesthetic model is represented as a normalized vector of contributing visual properties, with negative weights being displayed under the Y axis. The model's expected best spaceship and the participant's selected spaceship are also included for each iteration — the effect of the selection is visible in the next iteration's aesthetic model.

To evaluate the effects of the aesthetic model's adaptation, the Kendall rank correlation coefficient (Kendall's τ) will be used to measure the association between the initial presented spaceships (Fig. 13) and the same spaceships evaluated with each user's aesthetic model after 10 iterations (in the same order). The initial spaceships are used for reference — being the only content common among all participants — while Kendall's τ coefficient provides a measure of similarity of the fitness scores' orderings [43]. Its value ranges from 1 (if the orderings are identical) to -1 (if one ordering is the reverse of the other). Figure 16 displays each participant's τ coefficient values, accompanied with the final aesthetic model's best spaceship among those evolved after 10 iterations. Through the τ values we observe both substantial deviations from the initial ordering but also rather large differences among participants. Unexpectedly these results showcase the difference in visual taste and aesthetic values across different users, but also that very different expected best spaceships have similar τ values. The adaptation of the aesthetic score greatly depends on the presented content as well as previous selections (which cannot be illustrated due to space constraints); however, some observations can still be made from Fig. 16. Overall, many of the participants' models favored more complex shapes: such shapes usually have a negative τ score which corresponds to a large degree of reordering of the initial spaceships. The reordering is necessary since the initial aesthetic model favors simplicity (f_6 has an initial weight of 1) while the final aesthetic models usually do not — save for participants No. 10 and 11. On the other hand, a relatively simple spaceship such as the one for participant No. 5 lacks horizontal symmetry (f_1), weight concentration in the bottom (f_2) and weight concentration in the middle along the X axis (f_3): since all of these visual properties initially have a positive weight, in order for such a spaceship to be the best then the aesthetic model must have changed significantly, which is reflected in the participant's negative τ score. There are many subtle reasons why the adapted aesthetic models of different participants evaluate the illustrated spaceships as the best, many of which are hidden in previous selections or the successes and failures of previous evolution of new content. An overall conclusion however is that in most cases the "default" aesthetic model provided by the designer was significantly changed by the participants, resulting in a much more diverse and interesting variety of content.

Data pertaining to the online evolution of new content is documented in Table IV. The increase of the maximum fitness score during the short evolutionary sprints is small on average (0.037), with the average fitness score faring even worse. While the adaptation process of the aesthetic model is capable of finding the most appropriate content in the population and at reordering it to the user's preference, the optimization process rarely creates much better content within a single iteration. The

TABLE IV: Online evolution statistics collected from the user test. Mean values presented are calculated across all participants; standard deviation values are included in parentheses.

Presented spaceships' fitness range	0.49	(0.15)
Feasible population size	17.41	(4.25)
Improvement of max F within an iteration	0.037	(0.043)
Improvement of average F within an iteration	0.049	(0.096)

fact that optimization only runs for 5 generations at a time is one reason for these misgivings; another reason is the fact that from the 46 feasible individuals in the initial population, only 17.41 on average remain feasible on the ensuing evolutionary runs. The high dimensionality (given the small population) of the aesthetic score is the most significant factor for the slow optimization. Even in offline experiments presented in Section V-B with much larger populations, changes within 10 or 20 generations for an all-inclusive aesthetic model (Fig. 12c) are not much higher than the ones indicated here. The weights of the aesthetic models in the current experiment are different and possibly better suited for guiding evolution, but high-dimensional data always adversely affect the optimization progress. Although not reflected in Table IV, an inspection of the generated or presented content shows that the short evolutionary sprints between each iteration did in fact create novel (if not always better) content. Another insight gained from inspecting the content is the fact that the population tends to lose its diversity in later iterations due to convergence, especially if the user makes "consistent" choices. The result is a set of similar spaceships being presented to the user (such as the first four spaceships in Fig. 14) which necessitates a rather uninteresting choice on their part. Selecting no spaceship at all often mitigated such situations, but its effect also depends on the heterogeneity of the population and its ability to create very different content within the few generations it evolves. As a whole, the large fitness difference between the best and the worst presented spaceship (identified as *fitness range*, which was 0.49 on average) denotes that in most cases the user was presented with suitably diverse spaceships.

VI. Discussion

The adaptation of the aesthetic model based on the visual properties' score difference between selected and mean of unselected individuals had quite satisfactory results. However, it could be claimed that some aspects of the method are rather arbitrary, as some design choices were done based on experimentation rather than principle. For example, it could be that instead of a linear model, a non-linear measure of "difference" could better evaluate the difference between very similar spaceships (ignoring outliers) or focus on the differences between the selected spaceship and the most different unselected ones. Additionally, the adjustment of the aesthetic model when no spaceship is selected — combined with the arbitrary choice of 100 additional weight updates regardless of player choice — is largely based on intuition and preliminary tests. Future work needs to explore alternatives to these design choices, and document the benefits and drawbacks of each adjustment strategy empirically.

The presented adaptive aesthetic model consists of seven different visual properties. The addition of more visual properties would add granularity to the aesthetic model and help it identify many more factors behind a player's choice. However, optimizing a more complex aesthetic model would suffer from the curse of dimensionality [44] which, in part, also affects the content optimization progress in the presented user test. Previous user tests [4] tackled the problem by optimizing the two most "different" visual properties during each iteration. It is important to note that our suggested approach requires fitness functions which must be carefully designed by the creator. While the aesthetic model can adjust the impact of each of these fitness functions, the creator still assumes that the inserted visual properties are inclusive and representative of those of the public. Future work will explore the possibilities of using principal component analysis to identify wanted or unwanted patterns among the presented spaceships.

In addition to the types of visual properties being optimized (and the method they are computed), the core innovation of this paper over previous work [4] is the shift in representation. The enforced horizontal symmetry, as well as the attachment of weapons and thrusters to the top and bottom of the spaceship respectively, considerably helped users identify the generated content as spaceships. Unlike previous user tests, participants had no problem identifying the presented content as a spaceship and often went as far as visualizing its potential combat tactics or in-game uses. In cases where in-game performance is being optimized, the shift in representation is expected to significantly increase both the chances of individuals' feasibility and their fitness compared to previous approaches [5] — such implications will be explored in future work.

However, despite the changes in representation, some of the final generated spaceships shown in Section V do not appear as appealing as they could potentially be. We believe that this is chiefly due to that the spaceship hulls are represented as 2D polygons and rendered directly as such when presented to the user. The simplicity of the 2D polygon allows for the computation of various aesthetic measures with a small computational overhead and limits the number of constraints that need to be satisfied. But this simplicity comes at a price, since the final generated spaceships also appear plain; should the same results be rendered differently (without changing the underlying optimization strategy), the response from the players would likely improve. Future work will explore the possibilities of applying a procedurally generated texture with additional details over the generated polygon or reconstruct a 3D model using the 2D polygon as the base shape (akin to a slice image). Fig. 17 demonstrates how a generated spaceship can be rendered in a different fashion (in 2D or 3D) in order to improve its appearance.

Still, the result quality could conceivably be improved through further development of both the evaluation function and the representation, the two key considerations for search-based PCG [12]. Regarding the evaluation function, an unbalanced weighting of aesthetics can easily lead to less than stellar results; for example, optimizing for the simplicity aesthetic often results in rectangular or oval spaceships which are hardly visually interesting. Given that evolutionary algorithms

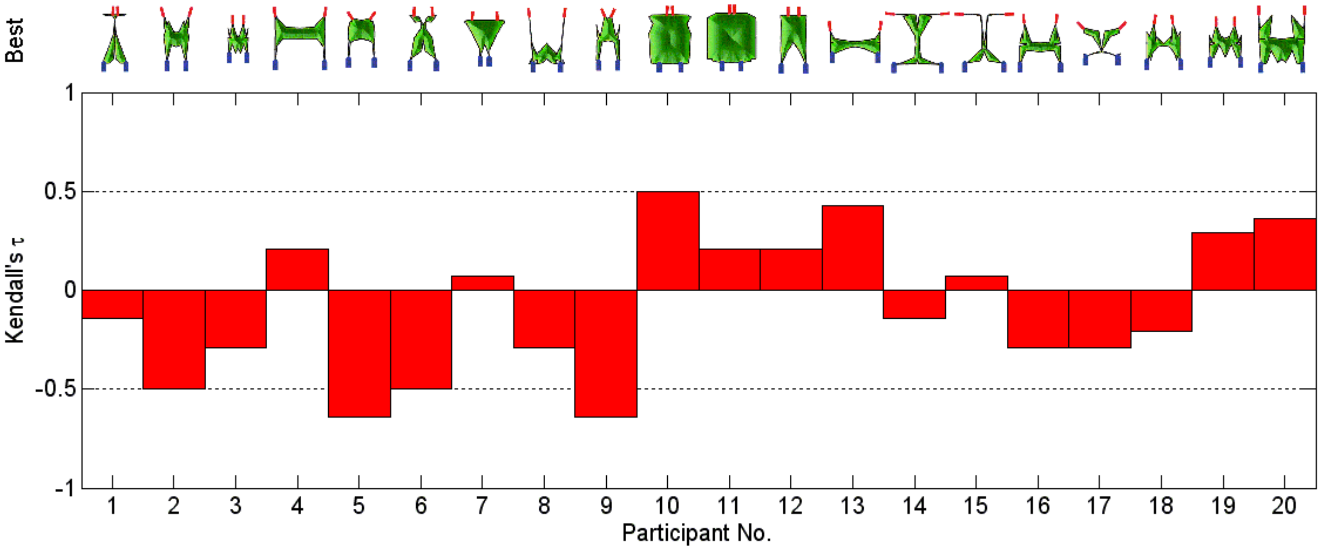
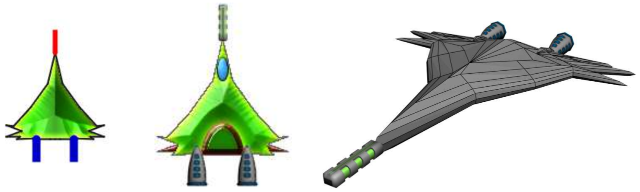


Fig. 16: Kendall's τ coefficient between the initial presented spaceships and the same spaceships evaluated with each participant's aesthetic model after 10 iterations. The final aesthetic model's best spaceship among those evolved after 10 iterations is also included for each participant.



(a) Base shape (b) 2D interpretation (c) 3D interpretation

Fig. 17: Artistic interpretations of spaceships optimized through the algorithm. The base shape (a) is the best spaceship optimized for f_5 from Fig. 10.

often exploit any shortcoming of the fitness function, all of the fitness functions which quantify different visual patterns may be responsible for certain spaceships' appearance. Future work should refine these evaluation strategies to limit their shortcomings.

The content representation affords several crucial design decisions with a strong impact on the possibility of finding good and diverse spaceships; core considerations include dimensionality, locality, completeness, and directness [12]. This work has used CPPNs as representation for spaceship hulls. Several other representations were considered, including shape grammars [30], turtle graphics [45] and combinations of primitive shapes [46]. However, based on previous experience shape grammars and turtle graphics were considered to have lower locality than CPPNs (small changes to the genotype are likely to result in large and unpredictable changes to the genotype and therefore to fitness), and shape grammars and combinations of primitive shapes were considered to have lower completeness (the search space comprises a smaller part of the space of all interesting spaceship hulls). An example of the superior locality of CPPNs is their ability to retain their underlying visual patterns while adding detail through the addition of new nodes — this is particularly useful for the iterative adjustment of both the generated content and the aesthetic model which evaluates them. A final, not insignificant advantage of CPPNs is that it has limitless resolution (this property is shared with fractals such as the Mandelbrot set, which however have abysmal locality and completeness properties for most types of content). It is of course possible and plausible that some hitherto unexplored representation (perhaps a variation on those presented previously) will prove to be superior to CPPNs in future work.

It should be pointed out that settling on CPPNs does not mean that there are no design decisions left for the representation. In particular the choice of activation functions for the CPPNs directly define the search space and therefore indirectly the spaceships' appearance; future work could explore alternative activation functions which may yield more visually interesting results.

The framework and algorithm proposed is applicable beyond the scope of spaceship design. The visual properties presented in this paper can be applied to any 2D game element, while 3D game content can also be generated and evaluated with small changes to the representation and the methods for calculating the visual properties' fitness scores. Additionally, the evaluation of content quality can be used as an authoring tool in game development: while the neuroevolutionary constrained optimization of multiple visual properties certainly has room for improvement, the adaptive aesthetic model in the form of a weighted sum can also be useful for evaluating existing content. This aesthetic model can select the most suitable game content for a particular player's visual taste from a collection of pre-generated or procedurally generated content. The player's choices can be used to adapt the aesthetic model much like the experiment presented in this paper and without the need for online evolution of new content. The fact that such a personalized aesthetic model can be combined with the evaluation of competencies in game-specific tasks [5], allows for a personalized experience where the presented content is to the player's taste and its functionalities are tailored to their playing style or challenge level.

VII. Conclusion

This paper introduces a generic two-step adaptation framework for the generation of personalized content with regards to a user's visual taste. In the proposed adaptation scheme the content is not only adapted to maximize a set of fitness functions, but the fitness functions that assess the visual quality of the content are themselves adjusted to match the aesthetic preferences of users.

To showcase the effectiveness of the proposed framework the paper presents a constrained search-based procedural content generation method to optimize the shape of spaceships. The fitness dimensions are based on visual properties deemed significant and "universal" by studies on human perception. By ensuring that candidate spaceships meet constraints imposed by the game engine and a human designer, the content generator operates in a much smaller and more relevant search space. The visual properties presented in this paper are only a sample of possible quantifiable aesthetic and functional characteristics, some of which have been introduced in previous work [41], pertaining to size, cost, or weapon- and thruster-specific properties. These aesthetic evaluations can be used in conjunction with evaluations of the generated content's performance in game-specific tasks, as presented in previous work [5], to create both functional and visually appealing spaceships.

Experiments in the optimization of a single visual property have as a whole shown promising results, especially compared to similar neuroevolutionary approaches. On the other hand, the aggregation of multiple visual properties in a single fitness score (as a weighted sum) is not guaranteed to generate content with all the required visual patterns since results heavily depend on the type of visual properties being combined.

The approach proposed and methods presented in this paper could very conceivably be applied to other content generation problems where human evaluation is in short supply or expensive.

Acknowledgment

Thanks to all the participants of the interactive evolution experiement. The research was supported, in part, by the FP7 ICT project SIREN (project no: 258453) and by the Danish Research Agency, Ministry of Science, Technology and Innovation project AGameComIn; project number: 274-09-0083.

References

[1] E. J. Hastings, R. K. Guha, and K. O. Stanley, "Evolving content in the galactic arms race video game," *CIG'09: Proceedings of the 5th international conference on Computational Intelligence and Games*, pp. 241–248, 2009.

[2] J. Togelius, R. De Nardi, and S. Lucas, "Towards automatic personalised content creation for racing games," 2007, pp. 252–259.

[3] C. Browne and F. Maire, "Evolutionary game design," *Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 1–16, 2010.

[4] A. Liapis, G. N. Yannakakis, and J. Togelius, "Optimizing visual properties of game content through neuroevolution," in *Artificial Intelligence for Interactive Digital Entertainment Conference*, 2011.

[5] ——, "Neuroevolutionary constrained optimization for content creation," in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, 2011, pp. 71 –78.

[6] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 99, no. PrePrints, 2011.

[7] Blizzard North, "Diablo," 1997.

[8] Gearbox Software, "Borderlands," 2009.

[9] MicroProse, "Sid Meier's Civilization," 1991.

[10] Interactive Data Visualization, Inc. (IDV), "SpeedTree," 2011, Year refers to the latest software release as of November 7, 2011.

[11] S. Schmitt, "World Machine," 1992.

[12] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation," in *Proceedings of the EvoStar Conference*. Springer-Verlag, April 2010.

[13] G. N. Yannakakis and J. Hallam, "Towards optimizing entertainment in computer games," *Applied Artificial Intelligence*, vol. 21, pp. 933–971, 2007.

[14] G. N. Yannakakis, H. P. Martínez, and A. Jhala, "Towards affective camera control in games," *User Modeling and User-Adapted Interaction*, vol. 20, pp. 313–340, 2010.

[15] L. Kessous, G. Castellano, and G. Caridakis, "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis," *Journal on Multimodal User Interfaces*, vol. 3, pp. 33–48, 2010.

[16] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, sep 2001, invited Paper.

[17] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley, "Picbreeder: evolving pictures collaboratively online," in *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2008, pp. 1759–1768.

[18] D. Rowland and F. Biocca, "Evolutionary co-operative design between human and computer: implementation of "the genetic sculpture park"," in *VRML '00: Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*. New York, NY, USA: ACM, 2000, pp. 75–79.

[19] J. Clune and H. Lipson, "Evolving three-dimensional objects with a generative encoding inspired by developmental biology," in *Proceedings of the European Conference on Artificial Life*, 2011, pp. 144–148.

[20] A. Hoover, M. Rosario, and K. Stanley, "Scaffolding for interactively evolving novel drum tracks for existing songs," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2008, vol. 4974, pp. 412–422.

[21] A. Martin, A. Lim, S. Colton, and C. Browne, "Evolving 3d buildings for the prototype video game subversion," in *EvoGames Workshop*, November 2010, pp. 111–120.

[22] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Interactive evolution for the procedural generation of tracks in a high-end racing game," *Interface*, pp. 395–402, 2011.

[23] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling player experience for content creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 54–67, 2010.

[24] G. N. Yannakakis, H. P. Martínez, and A. Jhala, "Towards affective camera control in games," *User Modeling and User-Adapted Interaction*, vol. 20, no. 4, pp. 313–340, 2010.

[25] R. Arnheim, *Art and visual perception: a psychology of the creative eye*, revised and expanded ed. (2004) ed. University of California Press, 2004.

[26] V. S. Ramachandran and W. Hirstein, "The science of art: a neurological theory of aesthetic experience," *Journal of consciousness Studies*, vol. 6, pp. 15–51, 1999.

[27] A. Smith and M. Mateas, "Answer set programming for procedural content generation: A design space approach," *IEEE Transactions on Computational Intelligence and AI in Games*, 2011.

[28] K. Sims, "Artificial evolution for computer graphics," in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '91. New York, NY, USA: ACM, 1991, pp. 319–328.

[29] D. Ashlock, S. Gent, and K. Bryden, "Embryogenesis of artificial landscapes," in *Design by Evolution*, ser. Natural Computing Series. Springer Berlin Heidelberg, 2008, pp. 203–221.

[30] G. Ochoa, “On genetic algorithms and lindenmayer systems,” in *Parallel Problem Solving from Nature – PPSN V*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998, vol. 1498, pp. 335–344.

[31] P. Machado and A. Cardoso, “All the truth about NEvAr,” *Applied Intelligence, Special Issue on Creative Systems*, vol. 16, no. 2, pp. 101–119, 2002.

[32] S. Wannarumon, “An aesthetics driven approach to jewelry design,” *Computer-Aided Design and Applications*, vol. 7, no. 4, pp. 489–503, 2010.

[33] S. Baluja, D. Pomerleau, and T. Jochem, “Towards automated artificial evolution for computer-generated images,” *Connection Science*, vol. 6, no. 2 & 3, pp. 325–354, 1994.

[34] P. Machado, J. Romero, A. Cardoso, and A. Santos, “Partially interactive evolutionary artists,” *New Generation Computing – Special Issue on Interactive Evolutionary Computation*, vol. 23, no. 42, pp. 143–155, 2005.

[35] K. O. Stanley, “Exploiting regularity without development,” in *Proceedings of the AAAI Fall Symposium on Developmental Systems*. Menlo Park, CA: AAAI Press, 2006.

[36] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[37] Z. Michalewicz, “A survey of constraint handling techniques in evolutionary computation methods,” in *Proceedings of the 4th Annual Conference on Evolutionary Programming*. MIT Press, 1995, pp. 135–155.

[38] M. Schoenauer and Z. Michalewicz, “Evolutionary computation at the edge of feasibility,” in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1996, pp. 245–254.

[39] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a feasible-infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, October 2008.

[40] E. Fogel, R. Wein, B. Zukerman, and D. Halperin, “2D regularized Boolean set-operations,” in *CGAL User and Reference Manual*, 3.9 ed. CGAL Editorial Board, 2011.

[41] A. Liapis, “Optimizing game elements based on aesthetic and performance principles and affected by player preferences,” MSc thesis, IT University of Copenhagen, Copenhagen, Denmark, February 2011.

[42] B. Widrow and S. D. Stearns, *Adaptive signal processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1985.

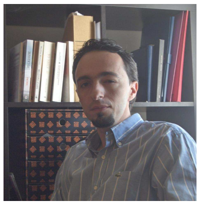
[43] D. Wilkie, “Pictorial representation of kendall’s rank correlation coefficient,” *Teaching Statistics*, vol. 2, no. 3, pp. 76–78, 1980.

[44] Bellman, R.E. and Rand Corporation, *Dynamic programming*, ser. Rand Corporation research study. Princeton University Press, 1957.

[45] G. Hornby and J. Pollack, “The advantages of generative grammatical encodings for physical design,” in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1, 2001, pp. 600–607.

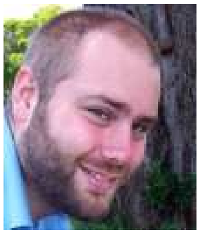
[46] A. Martin, A. Lim, S. Colton, and C. Browne, “Evolving 3d buildings for the prototype video game subversion,” in *EvoApplications (1)*, 2010, pp. 111–120.

Antonios Liapis is a PhD fellow at the IT University of Copenhagen. He received his 5-year Diploma in Electrical and Computer Engineering from the National Technical University of Athens in 2007 and the M.Sc. degree in Information Technology from the IT University of Copenhagen in 2011. His research interests include the mixed-initiative design of game content, procedural content generation, digital aesthetics and evolutionary computation.



Georgios Yannakakis is an Associate Professor at the IT University of Copenhagen. He received both the 5-year Diploma (1999) in Production Engineering and Management and the M.Sc. (2001) degree in Financial Engineering from the Technical University of Crete and the Ph.D. degree in Informatics from the University of Edinburgh in 2005. Prior to joining the Center for Computer Games Research, ITU, in 2007, he was a postdoctoral researcher at the Mærsk Mc-Kinney Møller Institute, University of Southern Denmark.

His research interests include user modeling, neuro-evolution, computational intelligence in computer games, cognitive modeling and affective computing, emergent cooperation and artificial life. He has published over 90 journal and international conference papers in the aforementioned fields. He is an Associate Editor of the IEEE Transactions on Affective Computing and the IEEE Transactions on Computational Intelligence and AI in Games, and the chair of the IEEE CIS Task Force on Player Satisfaction Modeling.



Julian Togelius is an Assistant Professor at the IT University of Copenhagen (ITU). He received a BA in Philosophy from Lund University in 2002, an MSc in Evolutionary and Adaptive Systems from University of Sussex in 2003 and a PhD in Computer Science from University of Essex in 2007. Before joining the ITU in 2009 he was a postdoctoral researcher at IDSIA in Lugano.

His research interests include applications of computational intelligence in games, procedural content generation, automatic game design, evolutionary computation and reinforcement learning; he has around 60 papers in journals and conferences about these topics. He is an Associate Editor of IEEE TCIAIG and the current chair of the IEEE CIS Technical Committee on Games.