Human Computation and Serious Games: Papers from the 2012 AIIDE Joint Workshop
AAAI Technical Report WS-12-17

Limitations of Choice-Based Interactive Evolution for Game Level Design

Antonios Liapis, Georgios N. Yannakakis, Julian Togelius
Center for Computer Games Research
Rued Langaards Vej 7
Copenhagen, Denmark

Abstract

This paper presents a tool geared towards the collaboration of a human and an artificial designer for the creation of game content. The framework combines procedural content generation using stochastic search with user input in the form of an initial goal statement as well as preference of generated results. Feedback from industry experts in a pilot user experiment showcased the limitations of this approach and the protocol chosen for evaluating the authoring tool. The limitations are discussed with respect to the suitability of interactive evolution for creative design and the design of experimental protocols for evaluating authoring tools for games.

Game content has often been generated algorithmically both in the industry and in academia, but many game designers are skeptical of the randomness inherent in generators (Champandard 2012). We argue that human-based computation is a viable alternative for the design of game content, using an intelligent tool to automate the mechanizable parts of content creation (such as playability checking) and suggest alternatives to handcrafted designs. With such a tool, human designers should be able to a) create high-quality content faster and with less effort and b) enhance their own creativity through the suggestion of novel alternatives to their designs.

This paper presents a first step towards actualizing such a tool, using strategy maps as a testbed and integrating human-based computation in the form of choice-based interactive evolution carried out by a single expert user. Using the tool, a designer creates an initial map acting as a template to generate map variants and to estimate the creator's intentions. A user preference model is updated based on designer choices among presented alternatives, and is used to generate personalized maps. An experiment with game developers using the presented tool showcased the limitations of our approach and of the protocol chosen for evaluating authoring tools. Requiring the manual creation of a complete map increased temporal and personal investment in the initial map design, which led to the rejection of any alternatives suggested by the tool. Additionally, the gameplay and aesthetic properties appraised by the preference model were rather arbitrary and represented a subset of designer criteria, limiting the model's ability to infer and accommodate designer preferences.

With the negative result obtained in this study we hope to shed some light primarily to the discussion about appropriate user study protocols for evaluating authoring tools as well as to contribute to the discussion about the suitability of interactive evolution for game authoring tools.

Related Work

The game industry has been using procedurally generated content since the eighties, particularly for levels in strategy games such as *Civilization* (Microprose 1991) or *SimCity* (Maxis 1989). In academia, genetic algorithms are becoming a popular solution for optimizing generated game content (Hastings, Guha, and Stanley 2009; Sorenson, Pasquier, and DiPaola 2011; Liapis, Yannakakis, and Togelius 2011).

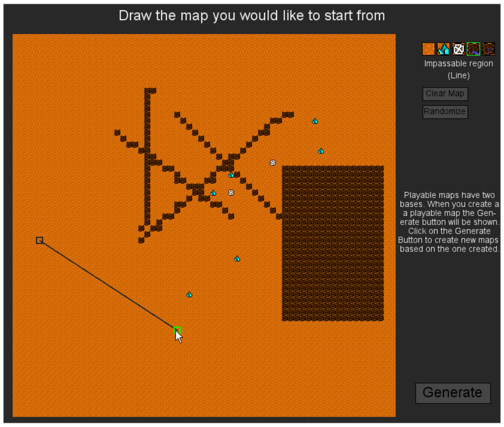
Contemporary game companies often generate content procedurally during development to reduce time and cost, with tools such as *SpeedTree*[1]. Such tools can afford to create content that is less constrained, assuming a human designer can detect flaws and discard bad content. Most such tools only allow designer intervention before (i.e. parameter setup) and after (i.e. approval or rejection) the generative step. However, a few tools (Smith, Whitehead, and Mateas 2011; Smelik et al. 2011) allow for computer and designer collaboration throughout the creative process.

Methodology

The maps generated by the system are abstractions of levels used in successful strategy games such as *Starcraft* (Blizzard 1998). All maps have a size of 64 by 64 tiles; tiles can be *passable* or *impassable*. Passable tiles can contain *player bases* and *resources*. The map layout assumes that each player starts at one of the bases and gathers resources to produce units; units move through passable tiles in order to attack the opponent's base. Each map is encoded in an array of real numbers within $[0, 1]$. Each player base or resource is encoded in 2 parameters, corresponding to the X and Y coordinate if multiplied by the map's width and height respectively. Each impassable region is defined by the coordinates of its diagonal corners and a parameter t defining the region as a rectangle ($t < 0.5$) or a line ($t \geq 0.5$).

A map editor was developed for this study to allow the manual creation of strategy maps by a human designer (see

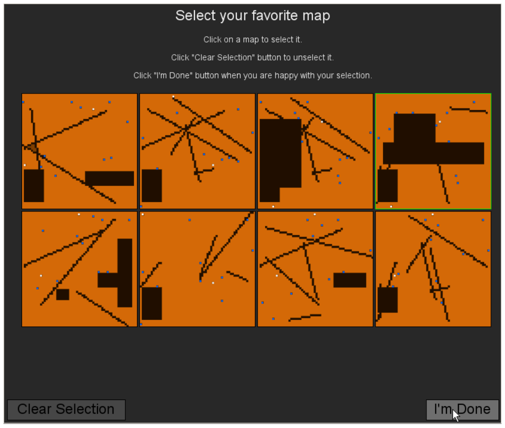
[1] http://www.speedtree.com



Draw the map you would like to start from

Playable maps have two bases. When you create a a playable map the Generate button will be shown. Click on the Generate Button to create new maps based on the one created.

Figure 1: A screenshot of the Map Editor window while a new map is being created manually.



Select your favorite map

Click on a map to select it.

Click "Clear Selection" button to unselect it.

Click "I'm Done" button when you are happy with your selection.

Figure 2: A screenshot of the Map Selection window while a user selects the favorite map among those presented.

Fig. 1). While the designer places features on the map (resources, bases, impassable areas), the map is tested for feasibility; once the created map satisfies all feasibility constraints, the designer is allowed to submit their map as the template map and refine it through interactive evolution.

For the purposes of this study, a map is feasible if it has two player bases and there are passable paths between every player base and every resource. Constrained optimization is achieved in this study by maintaining two populations (Kimbrough et al. 2008), one containing feasible maps and one containing infeasible maps. The population containing infeasible maps evolves towards minimizing the number of unreachable bases or resources. Both populations optimize their genotypes' parameters through mutation and recombination; parents are selected from within the same population using fitness-proportional roulette wheel selection.

The template map provided by the designer via the map editor contains information about a) the map's intended gameplay and aesthetic properties, but also b) the appearance of the map itself. Two respective steps are taken to preserve this information: a) several properties of the template map are evaluated and used for comparing generated maps with the template and b) the initial population of the genetic algorithm comprises of the template map itself and variations of it. The initial population (divided into feasible and infeasible) includes a copy of the template map and recombinations of the template map with random genotypes possessing the same number of each feature (bases, resources, impassable regions) as the template map.

For the purposes of this study, five gameplay and aesthetic properties of generated or hand-crafted maps are evaluated:

Base Distance which rewards a long distance between the two player bases:

$$f_{BD} = min\left\{1, d_B/(w_M + h_M)\right\} \tag{1}$$

Nearby Resource Balance which rewards a fair distribution of resources around each player's base:

$$f_{RB} = 1 - |r_{N,1} - r_{N,2}|/r \tag{2}$$

Impassable proportions which rewards a balance between impassable and passable areas:

$$f_{IP} = 1 - |1 - (2N_I)/(h_M w_M)| \tag{3}$$

Impassable Concentration (X) which rewards concentration of impassable tiles on the left half of the map:

$$f_{ICx} = N_{I,L}/N_I \tag{4}$$

Impassable Concentration (Y) which rewards concentration of impassable tiles on the top half of the map:

$$f_{ICy} = N_{I,T}/N_I \tag{5}$$

where d_B is the distance in tiles between the two player bases using an A* pathfinding algorithm, w_M and h_M are the map's width and height in tiles respectively, $r_{N,i}$ is the number of resources within a distance of 16 passable tiles from the base of player i, r is the number of all resources in the map, N_I is the number of impassable tiles in the map, $N_{I,T}$ and $N_{I,L}$ are the number of impassable tiles in the top half and left half of the map, respectively.

The hand-crafted map is evaluated via the above fitnesses: its scores are used to calculate the generated maps' similarity with the template map. Similarity s_i for fitness dimension i is calculated as $s_i = 1 - |f_i - f_{i_T}|$, where f_i and f_{i_T} the scores in fitness dimension i for the generated and the template map respectively. Combining similarity in each of the fitness dimensions into a weighted sum allows for the weights of this quality approximation to be adjusted in a straightforward fashion based on designer choices. The adaptive model can thus prioritize similarity with specific map properties over others; with negative weights, the model can also favor difference from the template map's properties. Because similarity uses an absolute difference, including the generated map's fitness scores (f_i) to the weighted sum allows the adaptive model to include designer intent towards increasing or decreasing scores in certain fitness dimensions. The adaptive model of designer preference in this study is identified as the preference score (F) and is calculated as:

$$F = \sum_{i=1}^{N} w_{s_i} s_i + \sum_{i=1}^{N} w_{f_i} f_i \tag{6}$$

where N is the number of fitness dimensions (five in this study), f_i is the fitness score of a fitness dimension i and w_{f_i} is its corresponding weight, s_i is the similarity of the fitness score of a fitness dimension i with the relevant fitness score of the template map and w_{s_i} is its corresponding weight.

The preference model presented above is used as the feasible population's fitness to guide content creation; it can be adjusted by *Choice-based Interactive Evolution* which involves a series of *iterations* of user interaction, weight adjustment and content evolution. In each iteration several feasible maps are presented to the user, who selects one favorite among them (see Fig. 2). This user choice informs the weight adjustments of the preference model, and the updated model is used as the fitness of the feasible population which evolves for a few generations. The adaptation method based on the selection of a single map as favorite has been successfully used for the generation of spaceships for a 2D game (Liapis, Yannakakis, and Togelius 2012); we expect that the inclusion of similarity scores will increase designer control over the generated artifacts. The initial preference model assumes that the designer only desires similarity with the template map: therefore initial weights w_{s_i} all start from $1/N$ while initial weights w_{f_i} start from 0. Treating similarity scores simply as additional fitness dimensions, the goal of the adaptive model is to reward properties with a higher fitness score in the chosen map compared to unselected ones and penalize properties with a lower fitness score in the chosen map compared to unselected ones. The weight of a fitness i when the user chooses map C is updated by:

$$\Delta w_{f_i} = \alpha(f_{i_C} - \bar{f}_{i_U}) \tag{7}$$

where α is a weight update step, f_{i_C} is the chosen map's score for fitness i and $\bar{f}_{i_U}$ is the average score for fitness i across all unselected maps. Eq. (7) is used for updating w_{s_i} scores, replacing f_{i_C} and $\bar{f}_{i_U}$ with s_{i_C} and $\bar{s}_{i_U}$, respectively.

The weights are adjusted until the chosen map has the highest preference score F among those presented; the adjustment process can be prematurely terminated if the preference score difference between the highest scoring map and the chosen map starts to increase or after $3 \cdot 10^5$ weight updates. Although with this adaptation method the preference model may "forget" previous selections and their inferred preferences, past preferences are retained in the population as it has been optimized from previous designer selections.

Experiment

In order to assess the potential of the proposed framework, five game developers were asked to use the tool in a pilot user survey. Participants included designers, game programmers and indie developers; they were male with ages between 26 and 33 and had experience playing strategy games. Participants were asked to design an initial map and select between presented maps for five iterations. Between iterations content was evolved for 10 generations according to the preference model of eq. (6), which was updated from the previous user selection. The participants' hand-crafted maps and those selected after five iterations are shown in Fig. 3. Results show that in most cases, designers chose generated content which matched the appearance of their hand-crafted designs the most. This resulted in high weights for similarity fitnesses and convergence of the population to the initial map's design. Although 40% of participants reported that choice-based interactive evolution improved their designs, all five designers selected the original hand-crafted map over the final selected one shown in Fig. 3.

Discussion

The presented tool's ambition was to allow more designer control over the generative algorithms via interactive evolution. Simultaneously, the automated content generation was expected to speed up the design of high-quality, playable content and enhance designer creativity through novel suggestions. The user experiment with a small sample of industry experts showcased the shortcomings of our proposed tool and the protocol used to evaluate it, since the requirement for a hand-authored initial map failed to speed up the design process, while the model's insufficient fitness dimensions failed to generate content conforming to designer taste.

The user study demonstrated that requiring the hand-authoring of a complete strategy map prior to any feedback from the system (other than feasibility checking) introduced a significant investment of time, effort and creativity from the human designer. Assuming that the human author had perfected their design during this initial stage, the alternatives proposed by interactive evolution were almost universally rejected, even if suggested maps were more optimal in one or more fitness dimensions. The experimental protocol we used thus failed to speed up content creation; limiting the designer's time investment in the initial map authoring stage should increase efficiency and reduce fixation over a specific design. Changing the map editor interface to allow for the creation of lower-resolution map sketches (using a map with fewer tiles) or for the description of broader goals (such as the number of nearby resources for each base) could reduce designer effort. On the other hand, integrating feedback from the system during the initial map creation, in the form of suggestions for alternatives and a visible evaluation of the map in different fitness dimensions may not reduce design time but may nevertheless enhance human creativity and allow for a more direct human-computer collaboration. Finally, allowing multiple designers to collaborate in the creative process by crowdsourcing content selection could enhance creativity and remove the implications of the initial creator's fixation; collaborative evolution of game content has demonstrated its potential at creating novel designs (Hastings, Guha, and Stanley 2009).

Observing the maps authored manually by industry experts (Fig. 3), symmetry was an important criterion behind many of these designs. However, symmetry is currently not among the five fitness dimensions on which content is evaluated. Verbal feedback from participants in the experiment on the importance of choke points or resource grouping highlighted that the fitness dimensions included in the model do not encompass the full range of designer criteria for high-quality maps. As the importance of these fitness dimensions is adapted through interactive evolution, a lot of information regarding designer preference is lost and the inferred preference model does not accurately match designer taste.

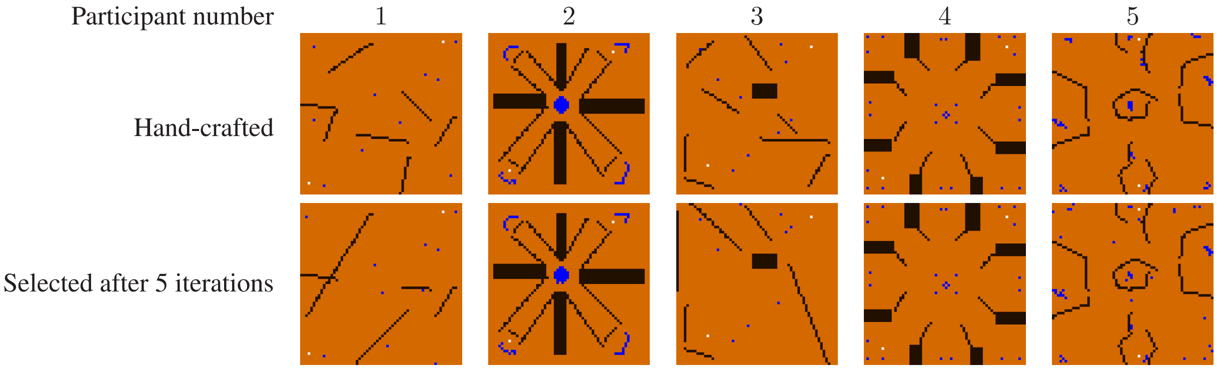


Figure 3: The hand-crafted and final selected maps of each participant in the user experiment. Dark areas represent impassable tiles, blue points are resources and white points are bases.

Additionally, the introduction of arbitrary parameters in the fitness evaluations (such as the cutoff distance for "nearby resources") adds bias towards specific map setups and limits the system's ability to generalize. Refining or increasing the fitness dimensions may confine the problem, but aggregating many fitnesses will hinder exploration of the search space. This problem may be solved by decoupling the preference model from the feasible fitness function; while the preference model can be used for the selection of maps to present, evolution of feasible content can be carried out without objectives, as novelty search (Lehman and Stanley 2011).

The current experimental protocol required users to create their own designs and compare them with the system's suggested alternatives. Results showed that, even with the ongoing personalization of generated maps via interactive evolution, computer generated content was not up to par to human designs. A less biased comparison could be between content generated procedurally with and without personalization. The game authoring tool can also be evaluted based on usability metrics, such as how often designers choose novel generated suggestions over their initial designs.

Despite the shortcomings of the map generator as an authoring tool, it can still be valuable as a completely automated content generator. The genetic algorithm can optimize maps for any of the provided fitnesses, creating playable maps which can be transferable to most strategy games.

Conclusion

This paper presented an authoring tool allowing for human-computer collaboration in the generation of strategy maps. The tool combines manual map creation, constrained optimization and interactive evolution in order to accommodate designer control while automating the mechanizable parts of content creation. However, the requirement for hand-authoring a complete initial map as suggested by our experimental protocol coupled with the limited strategy map features detected by the system resulted in the tool's failure to achieve its intended goals of increasing human creativity and speeding up content design. The shortcomings of the study pose important research questions for the suitability of choice-based interactive evolution as an authoring tool and illustrate the unforeseeable challenges one faces when designing evaluation experiments for authoring tools, at large.

Acknowledgment

This research was supported, in part, by the FP7 ICT project SIREN (project no: 258453) and by the FTP project AGameComIn (project no: 274-09-0083).

References

Champandard, A. 2012. EA investing in the next-generation with proceduralism post-SSX. Retrieved May 22, 2012, from http://aigamedev.com/open/teaser/investing-in-procedural/.

Hastings, E. J.; Guha, R. K.; and Stanley, K. O. 2009. Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games* 1(4):245–263.

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

Lehman, J., and Stanley, K. O. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* 19(2):189–223.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2011. Neuroevolutionary constrained optimization for content creation. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, 71–78.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2012. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games, Special Issue on Computational Aesthetics in Games*. (to appear).

Smelik, R. M.; Tutenel, T.; de Kraker, K. J.; and Bidarra, R. 2011. A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics* 35(2):352–363.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games* (99).

Sorenson, N.; Pasquier, P.; and DiPaola, S. 2011. A generic approach to challenge modeling for the procedural creation of video game levels. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):229–244.