Using Evolutionary Algorithms to Target Complexity Levels in Game Economies

Katja Rogers, Vincent Le Claire, Julian Frommel, Regan Mandryk, and Lennart E. Nacke

***Abstract*—Game economies (GEs) describe how resources in games are created, transformed, or exchanged: They underpin most games and exist in different complexities. Their complexity may directly impact player difficulty. Nevertheless, neither difficulty nor complexity adjustment has been explored for GEs. Moreover, there is a lack of knowledge about complexity in GEs, how to define or assess it, and how it can be employed by automated adjustment approaches in game development to target specific complexity. We present a proof-of-concept for using evolutionary algorithms to craft targeted complexity graphs to model GEs. In a technical evaluation, we tested our first working definition of complexity in GEs. We then evaluated player-perceived complexity in a city-building game prototype through a user study and confirmed the generated GEs' complexity in an online survey. Our approach toward reliably creating GEs of specific complexity can facilitate game development and player testing but also inform and ground research on player perception of GE complexity.**

***Index Terms*—Complexity, evolutionary algorithm (EA), game economy (GE), genetic programming.**

I. Introduction

ECONOMIC systems are the foundation of many games. They can materialize as economic challenges (stemming from moving resources either physically between places or conceptually between owners) or as rule sets governing "creation, consumption, and exchange of quantifiable" tangible (e.g., gold) or intangible (e.g., popularity) resources in internal economies [1]. Resource-based economic systems exist in genres ranging from first-person shooters (e.g., health points or ammunition) to racing games (e.g., in-game currencies to buy or upgrade cars). However, game economies (GEs) are particularly prominent in the city/base-building genre and resource management games, in which the player's primary challenge stems from producing and managing resources. For example, in *Factorio* [2], players begin with coal, copper ore, iron ore, and stone as raw resources. Next, players can produce subsequent resources via machines such as furnaces or refineries, often while using other resources as fuel. GEs can be elaborate systems, consisting of many resources and complex interactions between them [1], [3], and the design of GE parameters is likely to influence the game's complexity and difficulty. However, developing economies for games—or adjusting existing ones to achieve desired difficulty or progression—is complex, because game designers have little guidance on how to design GEs for a specific game complexity or difficulty.

The games research community has a history of supporting game designers to achieve specific goals through algorithmic tools. From procedurally generated levels [4], terrains [5], opponents [6], and narratives [7], through tailored game difficulty [8] and dynamic difficulty adjustment (DDA) [9], algorithms have been effectively used to support game design in a variety of genres. However—to the best of our knowledge-GEs have been omitted in this discourse. To facilitate game development and advance research into complexity preferences among players of infrastructure-building games, we suggest that procedural generation of GEs can be used to target and assess specific complexity levels. Our research aims to provide an initial exploration of how to assess complexity in GEs, how to generate them technically, and how to target specific complexities. This exploration will enable future work to investigate designed and perceived complexity within GEs, and how game difficulty and player experience is affected.

In this article, we present a technical prototype that uses evolutionary algorithms (EAs) to create GEs with targeted complexity as a proof of feasibility. Thus, we provide a first proposition—with our implementation—of what complexity might mean in the context of GEs. We performed a technical evaluation of the algorithms to compare different complexity measures and conducted a user study ($n = 28$) to examine perceived complexity in a prototype city-building game. Finally, we further validated the prototype's underlying GEs and investigated how complexity is perceived and preferred in infrastructure-building games through an online survey with 737 complete responses. Our contributions are as follows:

Manuscript received 10 June 2021; revised 17 December 2021 and 31 March 2022; accepted 3 May 2022. Date of publication 25 January 2023; date of current version 17 March 2023. This work was supported in part by the Institute of Media Informatics at Ulm University where the study was carried out, and in part by CIHR AAL. *(Corresponding author: Katja Rogers.)*

This work involved human subjects or animals in its research. The author(s) confirm(s) that all human/animal subject research procedures and protocols are exempt from review board approval.

Katja Rogers is with the Digital Interactions Lab, University of Amsterdam, 1012 WX Amsterdam, The Netherlands (e-mail: katja.s.rogers@gmail.com).

Vincent Le Claire is with the Databases and Information Systems, University of Augsburg, 86159 Augsburg, Germany (e-mail: vincent.leclaire@informatik.uni-augsburg.de).

Julian Frommel is with the Interaction/Multimedia Group, Utrecht University, 3584 CS, Utrecht, The Netherlands (e-mail: j.frommel@uu.nl).

Regan Mandryk is with the Department of Computer Science, University of Saskatchewan, S7N 5C9 Saskatchewan, Canada (e-mail: regan@cs.usask.ca).

Lennart E. Nacke is with the HCI Games Group, University of Waterloo, ON N2L 3G1, Waterloo, Canada (e-mail: len@uwaterloo.ca).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TG.2023.3238163.

Digital Object Identifier 10.1109/TG.2023.3238163

1) Implementation of a system that uses EAs to target specific complexity of GEs and a technical evaluation of the system-produced complexity.
2) A player study on perceived complexity when the generated GEs are deployed in a city-building game prototype.
3) An online survey of how complexity is generally perceived in infrastructure-building games and for the system-produced GEs specifically.

Our research presents a proof-of-concept for developing GEs with targeted complexity using EAs. A system that can reliably create GEs of specific complexity will facilitate game development, user testing, and player-customized gameplay. Further, this provides the foundation for systematic complexity research in GEs and aims to spark discussion on how complexity in GEs complements or contrasts with game difficulty.

II. Background and Related Work

Adams defined *GEs* as systems in which resources are moved between owners [1]. Based on this definition, any quantifiable game resource is part of a game's internal economy, from resources with object representation in the game to less tangible ones such as health points [10]. For this research, we focus on games that prominently feature the GE (i.e., infrastructure-building games). Generally, these games include elements that actively produce or spawn resources (e.g., mines) or resources that can be collected throughout the game environment (e.g., foraging mechanics). Other game elements allow players to craft new resources through the destruction or loss of other resources (e.g., creating copper plates from copper ore).

Despite the prevalence of GEs, there has been little research focusing on this formal game design aspect and its player experience effects. Lehdonvirta and Castronova [11] published a book on virtual economies, noting that the most common objectives of including economy design in a game are content creation, user attraction (e.g., via a free and a purchasable currency), and monetization. Drachen et al. have noted that GEs differ in complexity, operation, and design [3]. An earlier paper by Lehdonvirta [12] stated that GEs often get more complex than developers can easily oversee, describing the design of GEs as an "art [that] is still evolving." Dormans presented the machinations framework to describe the flow of game resources through diagrams that map the structure of GEs [13]. Following the machinations framework and the prior work of Adams, different game elements modify the number or ownership of game resources: Sources, drains, converters, and traders. *Sources* handle resource creation, *drains* their destruction, *converters* change resources into others, and *traders* swap resource ownerships. More recently, Stephens and Exton have explored inflation (and its mitigation) in virtual economies [14].

A. Difficulty and Complexity

GEs are inherently tied to game difficulty [1], [15]. When games feature a shortage of required resources or require a complex supply chain of resources to craft new ones, players must learn how to manage the resources they own and produce carefully. Particularly in complex GEs, shortages, or over-abundance of resources at any point along the supply chain can unbalance players' carefully managed virtual economy. The more involved management required from players by complex GEs thus results in higher difficulty.

In general, games try to provide progression in difficulty and challenge (i.e., through increasingly difficult game levels). This is commonly ascribed to flow theory [16], [17]. To reach flow in game design, a players' (increasing) skill level should roughly match the (increasing) challenge level afforded by the game. Doing this helps to avoid player boredom or frustration detrimental to player retention [18]. Much research has investigated difficulty adjustment effects on player experience [9], [19]. Player skill that roughly matches game challenges coincides with reduced frustration [20] and greater feelings of control and immersion [21]. These effects are linked to player enjoyment [22], [23]. Prior studies on difficulty adjustment occasionally yield outliers who feel frustration in low-challenge scenarios and boredom in the high-challenge scenarios [24]. These outliers are explained through individual player motivation differences [25]. Further, when game difficulty adjustments are not designed subtly, players can be dissatisfied by their achievements [26], [27].

Complexity in games has been explored from multiple angles. It has been presented as a continuum ranging from "casual" to requiring complex understanding, and shown to influence tutorial impact [28]. But specific types of complexity have also been considered, e.g., focusing on visual [29] or computational [30], [31] complexity. Further, Bowman has suggested that games can "vary in their cognitive, emotional, physical, and social complexity," thereby impacting how demanding they are [32]. To our knowledge, neither difficulty nor complexity adjustment has been applied to GEs.

B. Adjustment with EAs

There is an extensive academic precedence for the use of EAs with the goal of adapting game difficulty [7] (outside the purview of GEs). EAs begin with an initial population of candidates as solutions to an optimization problem (e.g., generating game levels that fulfill certain criteria) [33]. This population of candidates is then exposed to evolution as follows: The candidates are evaluated based on a fitness function (FF): In our example, fitness entails a certain degree of diversity and difficulty for the game level content. The candidates that achieve the worst result based on this evaluation are deleted and replaced by new candidates, which are generally produced through mutation and recombination of existing (well-ranked) candidates. The new set of candidates forms the next generation of the population.

Togelius et al. presented a taxonomy of the game elements for which evolutionary (and related) algorithms have been used for content generation; their survey showed that content generation has been applied in a wide variety of game content types, ranging from game rules, to game items and narratives [7]. Even after specifying our review of the literature to the use of EAs with the goal of difficulty adjustment, there are many examples. EAs have been used prominently toward evolving game strategies for more challenging and entertaining opponents and nonplayer characters [6], [34]. Game levels have been evolved for specific complexity [4], as have tracks in a racing game [8]. The closest relevant prior work has looked at procedural generation of game mechanics or rules [35], e.g., key elements like unit types [36]. However, despite the prevalence of GEs, neither difficulty nor complexity adjustment has yet been approached in the context of GEs, through EAs or otherwise.

III. Targeting Complexity in a City-Building Game

In this work, we aim to generate GEs through EAs. By designing these to target specific complexity, we automate the generation of GEs to facilitate (offline) authoring and testing (and better-suited complexity may also improve player experience). In addition to our EA that creates GE graphs of specified complexity, we thus implemented a city-building game prototype as a testbed.

A. Evolutionary Graph Generation

For our prototype, we consider GEs as graphs. In graphs, terminology consists of nodes (or vertices) and edges. GEs have the following two kinds of nodes: 1) *products*, which represent the resources existing in the virtual economy; 2) *producers*, which represent game elements that manufacture resources, i.e., they are connected to a product node via an edge. The edges of the GE thus describe the dynamic flow of resources within the GE. Producers can exist as *farms*, i.e., producing resources without the depletion of other resources (no input edges); or they can exist as *factories*, which use up resources to produce new ones (input as well as output edges). Based on this model, we designed and implemented an EA to create GE graphs with the goal of a specific complexity.

We chose genetic programming as the EA underlying our prototype graph generation, as it uses tree structures as representations for individual solutions (as opposed to binary or real number representation) [33]. This is well suited to generating graphs, as trees are a subset of graphs, i.e., undirected, connected, and acyclic graphs [37]. Further, to accommodate multiple end products as the game objective, we included a *needs* node as a root element to represent the final goal; we thus consider rooted trees as directed graphs [38]. GEs can easily be represented by a tree structure, with the minor constraint that they do not allow cycles within them.

1) Fitness Function: FFs drive the EA toward a specific goal by ranking individual solutions (i.e., in this case, GEs in tree form) of each evolutionary generation. This essentially defines what should be considered complexity in GEs. Given the lack of prior research regarding complexity and GEs, there were no indications for which attributes of GEs to target to author specific complexities. Intuitively, one might focus on the number of nodes of a given graph, or its depth. In a GE considered as a tree or graph, the depth would refer to the maximum number of edges from the root node to the graph's leaves. However, we aim to take into account aspects such as weighted edges, as well.

To do so, we propose the following FF (*Extended Recursive Weighted-Edges*, or *ERWE*) as a first working definition of complexity in a GE, expressed as positive integers: Complexity is calculated recursively over all nodes, but the function uses constants to weigh output amounts differently than input amounts. Products' complexity is defined as the complexity of their producer. Farms' complexity is the weighted output

$$c_{\text{out}} * o \tag{1}$$

with o as the number of resources produced by the farm per minute, and c_{out} as the constant used to weigh outputs. Factory complexity is defined as follows:

$$I_w + O_w + C_{\text{sum}} \tag{2}$$

where weighted input I_w is

$$I_w = c_{\text{in}} * (i - 1) \tag{3}$$

with i as the number of input resources per minute[1] and c_{in} as the constant used to weigh inputs. The weighted output O_w is defined in the same way as a farm's complexity, see (1). Further, C_{sum} is the sum complexity of all input nodes as recursively calculated using this method. The root node is treated as a factory node with no output. Based on pretesting, we used $\mathtt{c_{in}} = 2$ and $\mathtt{c_{out}} = 10$. The final fitness score is the difference between the calculated and target complexity.

The *ERWE* fitness function represents our first "best" approximation of complexity. Yet we do not assert this as the best measure of complexity for this scenario; instead, it is a rough estimate to explore the viability of this approach.

The following steps are applied onto an initial population of randomly generated trees (called individuals in EA terminology), until either a tree of the exact target complexity is reached, or a generation-based threshold is exceeded.

2) Parental Selection: After measuring fitness, the individuals are ranked by their fitness values. We tested the following four methods (chosen because they are simple and commonly used) to select parents for each generation from this ranking.

1) *Best*: The two individuals in each generation with the highest fitness (i.e., their complexity is closest to the target complexity) are selected as parents.
2) *Random*: Two individuals are chosen randomly with equal probabilities.
3) *Best and random*: The first individual has the highest fitness, the second is chosen randomly.
4) *Fitness proportional method*: Like a roulette wheel, this method [33] assigns to individuals a probability that corresponds to its fitness; individuals with higher fitness are more likely to be chosen.

This gave us a basic sampling of selection methods of various degrees of selection pressure. The *Best* method is essentially a truncation-style method (only the two highest-ranking individuals are chosen as parents), representing a variant with higher selection pressure [39]. *Random* in contrast represents a tournament-style variant with uniform probability distribution (any individual may be chosen), as a variant with lower selection pressure strength [39]. Both are also for example used in fitness approximate approaches to evolutionary computing [40]. The combined *Best and random*, and the *Fitness proportional method* represent variants with moderate selection pressure.

[1]The -1 in the weighted input I_w ensures that a minimal factory is not more complex than a minimal farm.

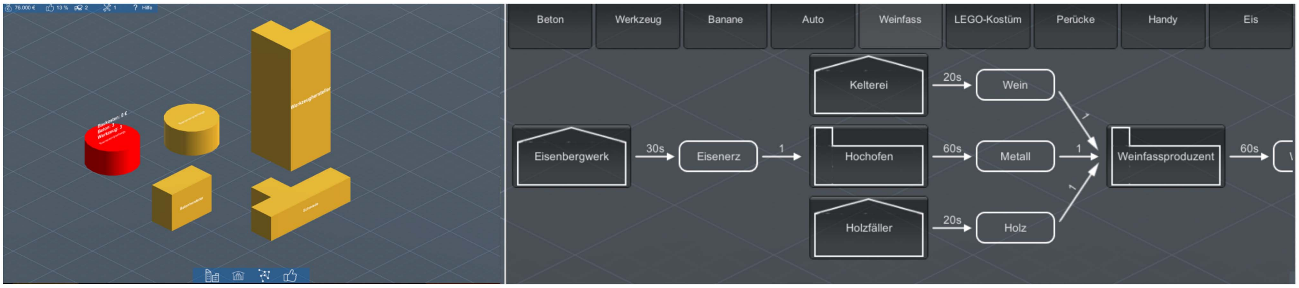


Fig. 1. Left: City-building game prototype *GenProcity* was developed for gameplay using the generated GEs. Right: *GenProcity* shows players the GE for each product (a gameplay video provided online: https://youtu.be/QWn-dIMOzkU).

3) Crossover, Mutation, and Replacement: Crossover operation was performed by randomly choosing a producer or product node of one parent through a reservoir sampling algorithm [41]. Then, a random node of the same type in the other parent is chosen with the same algorithm.[2] The selected node in the first parent is replaced by the subtree starting from the randomly chosen node in the second parent to create a new child tree. With a specific probability (5%), the EA also performs a mutation operation on the newly created child tree. This is applied by replacing a randomly chosen node of the child tree with a randomly created subtree (using the same random generation method used for the initial population).

A single new individual or child ("incremental" survival selection [39]) is thus created from the selected parents either by crossover, or crossover and mutation. It is then evaluated by the complexity function and compared to the existing trees. The worst individual (rated farthest from the target complexity) is then replaced by the new child tree if the latter has a better fitness ("replace-worst" replacement strategy [33]).

B. City-Building Game Prototype

In order to test evolutionary graph generation of GEs within an actual game, we required a simple infrastructure-building game. We implemented a game prototype called *GenProcity* (see Fig. 1), using Unity 5.3, to be played with a desktop computer and mouse. In the game, the player acts as a city planner with the objective of fulfilling the needs of the city's inhabitants by producing certain products. Given an initial capital, players spend their virtual currency to build factories and farms, in order to produce the required resources. Fig. 1 illustrates how players are informed of the individual GEs of each product in the game (i.e., a subset of the aggregated GE). Players essentially recreate the full GE generated at the game start until they either succeed to satisfy all inhabitant needs, or exceed the time limit (35 min).

A naming algorithm was implemented to assign names to products based on a knowledge base that contained information about names of products, and names of corresponding producers. The names are thus assigned based on the size of the individual subset GEs of end products, taking into account the number of input products and the number of producers.

[2]This selection always succeeds because even the minimal tree has one node of each type.

IV. Evaluation

The resulting prototype was subjected to a technical evaluation in order to compare the above described complexity functions (and parental selection method) in terms of runtime and accuracy. It should be noted that we did not aim to discover the best complexity function from this technical evaluation. Rather, we aimed to explore the viability of using the algorithm based on the *ERWE* FF for online in-game generation of GEs. *ERWE* represents a first attempt to define complexity in GEs in terms of tree width and height, while also incorporating input and output weights as well as input amounts. Yet, this can impact speed, and thus needs technical evaluation. If it performs poorly, more naive approaches (e.g., based on number of nodes, number of producers, or maximum depth) should be used due to the expected advantage in speed. Subsequently, we conducted a user study in order to investigate whether GEs developed to be of different complexity were also perceived as such by players.

A. Technical Evaluation

We compared the *ERWE* fitness/complexity functions with four parental selection methods for evolutionary generation of graphs targeting four different target complexities (1x4x4). Target complexity values were chosen to reflect the most simple possible GE (10), a moderately simple GE (30), a moderately complex GE (50), and a very complex GE (200). Each of these $1 \times 4 \times 4 = 16$ configurations were run 100 times, with an initial population size of 10. The number of products at the root node, and input edges for each factory were allowed to range from 1 to 4. The weight of each input, and the weight of each output were allowed to range from 1 to 3. A maximum threshold of 1000 generations was applied.

We focus our reporting on the results in terms of runtime (in milliseconds), and how often it found the exact target complexity. As shown in Fig. 2, the average EA run time with this FF ranged from 48.83 ms for the most simple target complexity, to 614.74 ms for the very complex target complexity.

The worst case runtime of the EA with the *ERWE* FF was roughly 600 ms, i.e., 0.6 s. This is slower than advised for systems aiming to ensure that users perceive system reactions as instantaneous (0.1 s), but still within the range for not unduly interrupting users' flow of thought (1 s) [42], [43]. It should thus support the generation of GEs within a duration acceptable

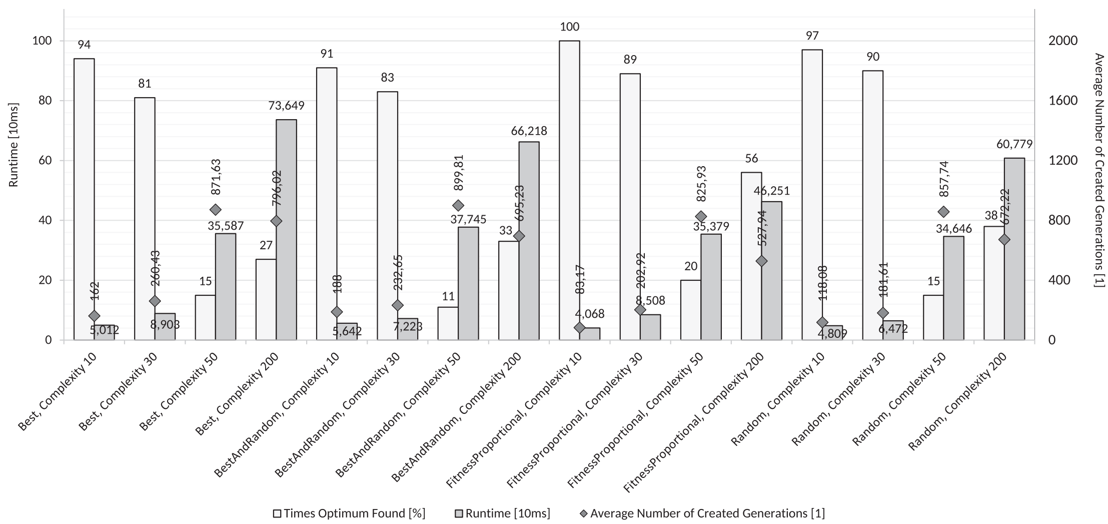


Fig. 2. Mean results for the *ERWE* complexity function in terms of runtime (left axis, in units of 10 ms), generations created (right axis, diamond notation with vertical label), and how often the optimal solution was found; the graph shows results for the four complexity values (10, 30, 50, 200), and all four parental selection methods.

TABLE I
GENERATED COMPLEXITIES WERE CONSISTENTLY VERY CLOSE TO THE TARGET FOR THE *ERWE* TEST RUNS

Target Complexity	10	30	50	200
Mean Best Final Complexity	10.69	30.32	51.60	200.77

for supporting rapid playtesting. In aiming for the exact target complexity, *ERWE* showed only modest success at first glance, getting progressively worse with higher complexity targets. For the simplest complexity target, 91%–100% of test runs found the exact target complexity of 10 (across the different choices for parental selection). For the moderately simple complexity (30), this ranged between 81% and 90%. This decreased to 11%–20% for the moderately complex target, and 27% –56% for the most complex target (a larger target may be more difficult). In the last two cases, the best percentages of exact solutions found (20% and 56%, respectively) occurred with *Fitness proportional method* parental selection, perhaps better avoiding local minima. Yet overall we note the average complexity value of all test runs' fittest final candidate produced via *ERWE*: The mean complexity of the final GEs is consistently very close (likely indistinguishably so) to the target, see Table I. Hence, while the data suggests that for higher complexities (50 and 200), the algorithm did not always reach the exact optimum, for our practical purposes the difference between target and final complexity is minor enough to likely be imperceptible. In summary, we found that genetic programming using our *ERWE* FF can successfully and within a reasonable time produce GEs of the targeted complexity.

B. *User Study: Perceived Complexity in Prototype*

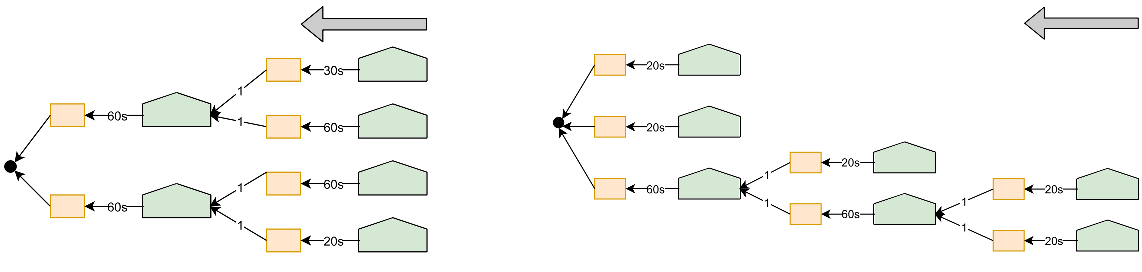
After the technical evaluation to determine the viability of using the EA for online generation of GEs, we conducted a user study to explore player perception of the measure of complexity defined through *ERWE* . Our goal was to investigate whether players actually perceived GEs generated with higher complexity values as more complex when embedded in gameplay, explored through a between-subject design.

1) Stimuli: Based on the technical evaluation, two graphs were generated using *ERWE* as the complexity function and *Best* as parental selection. One graph was generated for the target complexity of 100 (for the study context termed *low complexity*) and one for a target complexity of 175 (*high complexity*), based on pilot testing for average game durations. Example graphs are shown in Fig. 3, but they were generated in-game for each participant, i.e., the exact complexity value will have differed slightly. Gameplay was restricted to a maximum of 35 min, but could end earlier if players won before this (i.e., completed the full GE).

2) Measures: Given a lack of standardized questionnaires for complexity, we used custom seven-point Likert scale items to explore general enjoyment, and whether the GEs had been comprehensible, complex, and whether many steps or semifinished products had been required to create end products (for item wording, see Fig. 4). We also logged whether players won the game, i.e., managed to build the full GE to satisfy the inhabitants' needs within the allotted duration.

3) Participants and Procedure: Participants ($n = 28$) were recruited via bill-board postings, mailing lists, and several Facebook groups in a university setting (21 male, 7 female, 0 nonbinary). On average, participants were 23 years old ($M = 23.11$, $SD = 5.01$). Participants were divided evenly into the groups *low complexity* (11 male, 3 female) and *high complexity* (10 male, 4 female).

After consent procedures, participants were asked to fill out a pregame questionnaire assessing demographic background and gaming habits. They were then asked to play the game (with the GE set to their randomly assigned condition) for 35 min, or until they won. Subsequently, they were asked to fill out a



(a) (b)

Fig. 3. Examples (a) and (b) were generated with our EA, using *ERWE*. The graphs are "played" from right to left (direction of the gray arrow); players build farms and factories (green building shapes; farms have no input node) until they reach the end goal, i.e., the leftmost root node. Weighted edges indicate how often nodes are produced in a certain time frame, e.g., one unit of the right-most resources (orange rectangles) in (b) is produced every 20 s by the right-most farms. The calculation of the exact complexity for both examples (96 and 176, respectively) is illustrated in the supplementary materials. In the user study, comparable graphs were generated; these exact graphs were employed in the survey.

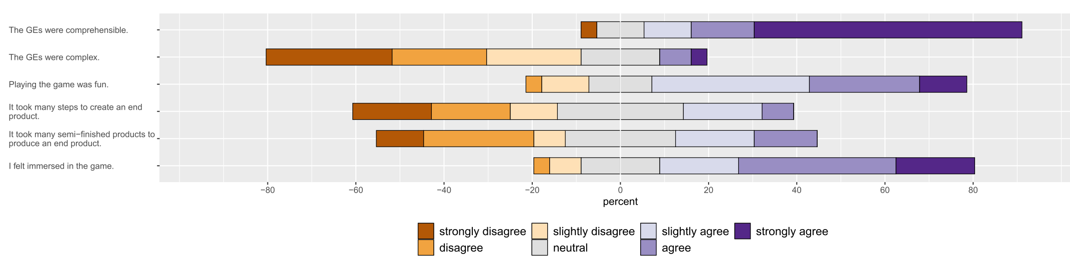


Fig. 4. Study participants' responses (here: overall) were largely positive, if indicating that the GEs were not overly complex. The statements regarding requirements for producing an end product differed significantly between conditions (see Fig. 5).

post-game questionnaire on their game experience. Remuneration consisted of 5 EUR and sweets. Each study session took between 45–60 min.

4) Results of User Study: Regarding gaming habits with infrastructure-building games, 39% of participants reported playing this kind of game at least once a month; 18% reportedly played at least once a week (detailed responses are in the supplementary materials). On a seven-point Likert scale (1 = *strongly disagree*, 7 = *strongly agree*), participants reported roughly neutral values for being experienced in the playing of infrastructure-building games ($Mdn = 4$, $IQR = 3$–5), but more positive values for enjoying playing them ($Mdn = 6$, $IQR = 4$–6.25). We present the overall results for participants across both conditions in Fig. 4. Participants in both conditions enjoyed the game and indicated that they felt immersed. Independent t tests revealed no significant differences between the two conditions for the question regarding enjoyment, or immersion. An independent t test and a Wilcoxon rank sum test indicated no significant difference in perceived complexity or comprehensibility between conditions, respectively. However, there was a significant difference for the statements that end products required many steps to produce, $t(24.91) = -2.16$, $p < 0.05$, $r = 40$. The ratings here were higher for the *high complexity* ($Mdn = 4$, $IQR = 3.25$–5) than the *low complexity* condition ($Mdn = 2.5$, $IQR = 2$–4). The same emerged for the *high complexity* ($Mdn = 4$, $IQR = 4$–5.75) versus *low complexity* condition ($Mdn = 2.5$, $IQR = 2$–4) with regards to end products requiring many semifinished products, $t(25.99) = -2.52$, $p < 0.05$, $r = 0.44$. This is illustrated in Fig. 5.

There was a clear difference in gameplay, 85.71% of participants playing with *low complexity* won, while only 14.29% of participants won in the *high complexity* condition. Binary logistic regression showed the complexity condition was a significant predictor of game outcome, χ^2 (1) = 10.01, $p < 0.001$.

5) User Study: Summary and Discussion: The results show no difference between the two groups in players' rating of the complexity of their GE. Nevertheless, the *high complexity* condition resulted in significantly higher agreement for the statement that many steps or semifinished products had been required to produce end products compared to the *low complexity* condition. Further, there was a significant difference in win rates, implying a greater difficulty inherent to the *high complexity* condition.

These two factors (the different win rates and perceived difference in steps required) suggest that the difference in complexity between the two GEs did persist when embedded in gameplay. However, it is possible that neither GE was considered highly complex, i.e., the graphs were simply too close in complexity values. It is possible that players would have determined a difference in complexity if confronted with a direct comparison of the GEs. However, it is also possible that the concept of complexity is difficult to grasp in the context of GEs, whereas "many steps" is more tangible.

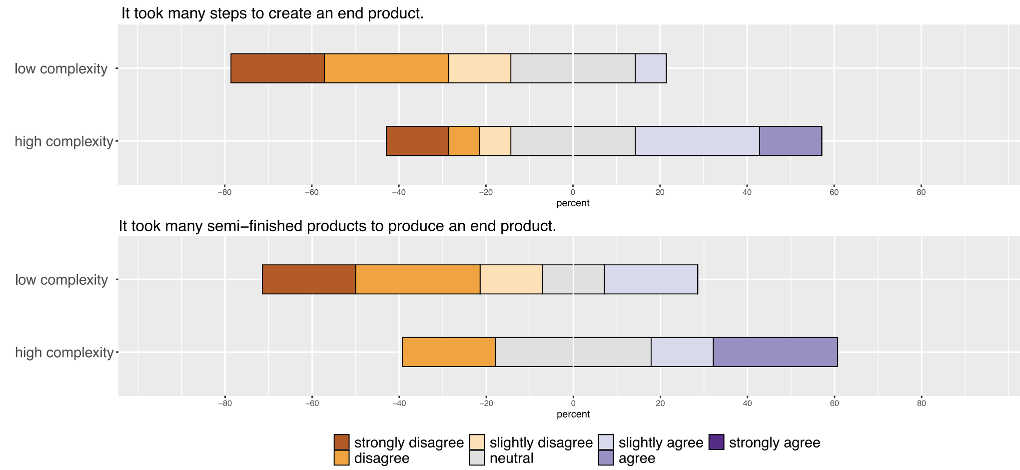


Fig. 5. GEs differed significantly in the number of steps and interim products required to create end products.

Finally, it should be noted that despite the difference in winning rates and the perceived difference in required steps, there was no difference in reported enjoyment.

V. Exploratory Online Survey

To explore our questions about complexity of GEs in a broader sense, and regarding our generated GEs in particular, we conducted an online survey via Reddit. With this survey, we investigated the following: How is the complexity of two comparable generated graphs (Fig. 3) perceived when both graphs are presented in comparison to each other? Further, we targeted information to situate the generated GEs in relation to those in similar existing commercial games: How complex are GEs in existing city-building games? With the goal of eventually being able to target specific existing games, how does complexity of the generated GEs compare to that of existing games?[3]

A. Measures

The survey began with questions to assess participants' demographic background, as well as their favorite game, and corresponding gaming habits. Most subsequent questions then referred to the entered favorite game.

Next, we used an example to introduce participants to the concept of GEs, and the terms producer and product. Given this example, we asked participants to confirm the existence of products and producers in their favorite game, and to name examples. This also functioned as a check to see whether participants had read and understood our introduction to GEs and their components.

We next showed participants two generated GEs (Fig. 3). We asked participants to rate their complexity (five-point Likert scale from 1 = *not at all complex* to 5 = *extremely complex*). We also asked whether they thought they would enjoy this complexity in gameplay (e.g., *I would enjoy the GE complexity in example A*, 1 = *strongly disagree*, 5 = *strongly agree*). Participants were then asked to rate how the GEs in their favorite game—on average, or the least and most complex occurring ones—compared to the example. Answer options here consisted of *less complex than A*/*more complex than B*, *roughly as complex as A/B*, and *more complex than A but less complex than B*. After this, we also asked participants to estimate their confidence regarding having understood the concept of GEs on a five-point Likert scale (1 = *strongly disagree*, 5 = *strongly agree*).

B. Survey Preparation: Targeting Subreddits

We prepared our survey by collecting a list of games with infrastructure-building mechanics, based on games tagged with *Building*, *City-Building*, and *Base-Building* on the Steam website.[4] Based on this list, we looked up subreddits with a thematic focus on these specific games,[5] as well as more general games-focused subreddits. Recruitment was always preceded by checking subreddit guidelines; the final list of subreddits in which the survey was promoted was constrained by the subreddit moderators' response and permission to post. If the moderators approved, we posted our survey link as a Reddit text post. A total of 19 subreddits allowed recruitment, consisting mainly of subreddits focusing on specific games, as well as one general subreddit (i.e., /r/BaseBuildingGames).

C. Participants and Procedure

We recruited participants from the 19 subreddits. Renumeration consisted of being invited to take part in a draw for one of five vouchers worth 25 USD on Amazon.com (or equivalent on.ca, .co.uk, or.de, subject to the participant's choice).

A total of 1517 responses were recorded; removal of incomplete datasets left us with 737 completed responses.[6] Participants were 26 years old on average (*IQR* = 21–30), and predominantly

[3]The survey also targeted players' game audio habits (omitted for scope).

[4]e.g., [Online]. Available: http://store.steampowered.com/search/?tags=7332

[5]We also contacted forum moderators (e.g., for *Factorio* [2], *Dwarf Fortress* [44], *Settlers* [45], and *SimCity* [46]) but received no response.

[6]Completed for the purposes of this survey; 20 participants opted out of supplying their email address for the draw.

TABLE II
COMPLEXITY VALUES AND MENTIONS IN PARENTHESIS ARE BASED ON POST-EXCLUSION RESPONSES

Favourite Game	Mentions	Complexity *Mdn*	Complexity *IQR*
Dwarf Fortress	165 (80)	3	3–4
Factorio	117 (78)	3	3–4
Cities: Skylines	96 (45)	1	1–1
Rimworld	90 (64)	2	2–4
Banished	53 (51)	2	2–3
Civilization series	40 (28)	2	1–2
SimCity series	39 (0)	–	–
Anno series	21 (18)	3	2.25–3
Stardew Valley	16 (12)	1	1–2.25
Prison Architect	14 (6)	1	1–1.75

male (634 participants, 86.02%). 75 participants (10.18%) identified as female, and 10 (1.36%) as nonbinary[7]. Participants ranged from a variety of 52 different countries: The majority were from the USA (42.20%), Germany (9.50%), U.K. (8.68%), or Canada (7.06%)—all other countries represented less than 3% of the participant sample. Reported employment status was similarly diverse: Most were employed in highly varied occupations (44.50%), were in school, training, or university (38.31%), or selfemployed (6.24%).

D. Results

We began by confirming that participants were confident in understanding the explanation of GEs ($Mdn = 4$, $IQR = 4$–5).

1) Favorite Games: Based on textual analysis, the favorite games listed were coded to group games of the same series, and correct typographical mistakes. Responses listing multiple games or games that could not be identified were removed from analysis. The remaining list consists of 42 different games. Table II lists the ten most commonly mentioned games along with their mean reported complexity.

2) Products and Producers: We used these questions in part as a manipulation check to the reported confidence in understanding GEs. 86.84% of participants confirmed the existence of products in their favorite game. The remainder denied the existence of products in their favorite games, even though the reported games clearly do have products (e.g., *Cities: Skylines* [47]), and were thus removed from analysis. Producers without input products, i.e., farms, reportedly existed in 68.25% of reported favorite games, and 78.83% of participants reported producers with one or more input products (i.e., factories). Again, the remainder incorrectly reported no such elements in their favorite game, and was assumed to have not properly read or not fully understood the introduction to GEs. After these additional exclusions, the dataset consisted of 455 responses; the following reports are based on this subset of the data.

3) Perceived Complexity in Favorite Games: Complexity values for participants' favorite game were roughly neutral ($Mdn = 3$, $IQR = 2$–3), with expected high values for enjoyment, $Mdn = 4.00$, $IQR = 4$–5. Kendall's tau showed a significant positive correlation between perceived complexity and enjoyment thereof, $z = 10.59$, $p < 0.001$, $r = 0.43$ (moderate effect).

4) Perceived Complexity of Generated GEs: Neither of the two GEs was perceived as particularly complex, however, the *low complexity* GE ($Mdn = 2$, $IQR = 1$–2) was perceived as less complex than the *high complexity* GE ($Mdn = 3$, $IQR = 2$–3). A Wilcoxon signed rank test showed that this difference was significant, $V = 3230.5$, $p < 0.001$, with an effect size of $r = 0.71$. In terms of expected enjoyment, both had the same median ($Mdn = 4$), but the *low complexity* version had a lower interquartile range ($IQR = 3$–4) than the *high complexity* version ($IQR = 4$–4). A Wilcoxon signed rank test showed that this difference was also significant, $V = 3610$ $p < 0.001$, albeit with a smaller effect size, $r = -0.28$.

5) Comparison of Complexity: In comparison to the complexity in the example GEs, participants were asked to consider and rate the average GE, as well as the least and most complex GEs occurring in their favorite game. In terms of average GE in their favorite game, most participants judged it to be either roughly as complex as the *low complexity* graph (31.65%) or less complex than the *low complexity* graph (24.52%). For the least complex GE in their games, 80% of participants judged this less complex than the *low complexity* graph. The most complex occurring GE, however, was estimated roughly as complex as the *high complexity* graph (17.58%) or more so (45.27%) by most participants.

E. Survey Discussion

The results show that the participants did perceive a difference in complexity between the two generated GEs. As such, while the difference between the two was not overly pronounced, the survey validates the results of our prior technical evaluation and user study: The EA appears to generate graphs that are broadly perceived as of different complexity.

Participants did indicate that they would enjoy both versions, but the *high complexity* version more so, although this difference was much less pronounced in terms of effect size. This supports the small positive correlation found between complexity and enjoyment of complexity in relation to participants' favourite game. However the smaller effect sizes also point toward the existence of players who enjoy GEs with lower perceived complexity, in line with related work on pottering and idle games [48], [49].

The results comparing the complexity of participants' favorite games provides a rough estimation of how complex average GEs in these games are, which is roughly or less complex than the generated *low complexity* graph. The large majority of participants also indicated that the least complex GE was less complex than the *low complexity* graph; presumably, this refers to tutorial levels or early stage GEs in participants' games. Finally, the most complex GEs in participants' games are judged to be as complex as the generated *high complexity* graph or more so; presumably this indicates progressing levels of difficulty within these games.

[7]This included transsexual or genderfluid gender identities, but also facetious answers like "*Dwarf*" and "*Yamamoto class battleship*."

VI. Overall Summary and Discussion

We focus our discussion on the viability of using an EA to generate GEs of specified complexity, our working definition of complexity, and finally lessons learned on how complexity might be perceived by players.

A. Technical Viability

Our implementation of an EA works to produce graphs, and the technical evaluation showed that the runtime for this graph creation is within roughly 0.6 s. As such, it could even be used for online generation within active gameplay, if this is conducted at loading time or between levels. While the exact target complexity was not always reached, the average final complexities are sufficiently close to this mark. Future work will have to explore optimization for the parameters of this EA (e.g., other fitness-proportionate selection methods like tournament [33]). Further, it should be noted that our current iteration does not support cycles within the graphs (in terms of gameplay, this could mean recycling resources via an earlier producer of the GE). This kind of feature is not possible without switching to a different type of EA (that does not use trees).

B. Defining Complexity

For this work, we used the *ERWE* FF, which worked well as a first example for measuring complexity. The function includes many parameters which could likely be optimized in future work to improve player experience of the resulting GEs. For example, the constants c_{in} and c_{out} currently more strongly weigh output products. In future work, we can explore the use of different values in actual gameplay, e.g., setting these constants to equally weigh number of input edges and output weights, or to emphasize the input edges. Further, it may be interesting to develop a different FF for GEs that is able to provide more nuanced aspects of complexity such as weighted edges or selection methods that avoid local minima. In general, however, the findings support the use of EAs in generating GEs for targeting specific complexity, using our working definition. This sets the foundation for automated adjustment of complexity in GEs, leaving further optimization for future work.

C. Perception of Complexity

The perception of complexity in GEs appears to be more difficult when there is no direct comparison, as in our user study. We suspect that this aspect may be easier for players when the graphs are obvious outliers, i.e., very complex or very simple. However, it is also possible that GEs need more pronounced complexity differences to enable such distinction. Especially when embedded in gameplay, players' perception of the GE may be altered or obscured by extraneous scaffolding in terms of teaching game tutorials, and planning next steps. Moreover, Denisova et al.'s work on perceived challenge has suggested a potential "invisible complexity" construct in games [50], that can derive from the management of game resources. Our results show that a difference was perceived in terms of low-level characteristics like required steps and interim products, but not in overall perceived game complexity. It is possible that our adjustments of GEs were thus part of this "invisible complexity."

We found a significant positive correlation between complexity and enjoyment in the online survey, which—assuming that higher complexity implies greater challenge—is in line with related work on enjoyment of challenge in games [16], [51], [52]. However, it should be noted that less complex games are also highly popular, as shown by related work and our online survey [48], [49]. Interestingly, regardless of perceived complexity, the generated GEs were perceived as equally enjoyable, both by players in the user study, and by participants rating their expectations in the online survey. Given the differences in perceived complexity of survey participants' favorite games (i.e., presumably, as considered favorites—and based on the ratings—enjoyable), we ascribe this enjoyment to the sample of players in the user study. Despite the found correlation between complexity and enjoyment, players' enjoyment of the games mentioned in the online survey were high even with low perceived complexity. It should be noted that these games incorporate more game mechanics than just building infrastructure; as such, players may place a greater importance on these other aspects (combat mechanics are also often found in these games, e.g., *Dwarf Fortress* [44] and *Stardew Valley* [53]). Yet in general, this shows that while many players greatly enjoy complex GEs, there is a noticeable number of popular games with simpler GEs. Whether this enjoyment is then still based on challenge outside of the GEs will be the subject of future work.

Overall, we conclude that, regardless of *perceived* complexity, the generated GEs did lead to a difference in terms of gameplay complexity (i.e., the number of steps required, and influencing the odds of winning or losing), and thus also achieved difficulty adjustment. Further, we note that most criticism of difficulty adjustment cautions against the use of overt modification to gameplay [26], [27]. With this approach of adjusting GE complexity, the difference was clearly subtle. It may thus be able to avoid players feeling as though their achievements mean less because the system helped them by providing an easier level.

D. Limitations

Our FF *ERWE* was derived from discussions and considerations between the first two authors, and can only claim to be a first working example of GE complexity. Other alternatives may prove more appropriate in future work. For example, we acknowledge that our representation of economic complexity does not cover concepts like supply/demand, competition between producers of the same resource, trading economies, or inflation. Some of these concepts could be represented through an adjusted FF, however others (e.g., requiring the above mentioned cycles) would necessitate switching away from genetic programming. Regardless, it suits our objective of showing initial technical feasibility. Further, both the user study and online survey seem to confirm that while the difference between the two GEs was not stark, it did exist in the sense that it was able to target different complexities.

For our user study, we note several limitations: Our sample size was not large, especially for a between-subjects design, and participants were mostly male. Future research could explore potential individual differences in complexity preferences, to provide a better picture of relevant factors. Moreover, we did not employ standardized questionnaires for complexity perception, as these do not yet exist. Such will be needed to reduce subjectivity in evaluating GE complexity. Further, comparing economies drawn from the games in Table II would be a great angle for in-game comparisons in future work.

Finally, our online survey on Reddit consisted of a strongly male leaning sample. We also note that despite our checks for participants being confident in their understanding of GEs, and being able to apply the concept to their favorite games, it is possible that some ill-intentioned answers remained in the sample and constitute bias (e.g., facetious responses). Generally, results must be considered in light of our comparison being between an abstract GE visualization and a GE embedded in a game. Additionally, participants' favorite games employ GEs to support gameplay differently, e.g., win conditions that do not relate to an end product. Future studies will have to be conducted for a more direct comparison.

VII. Conclusion

In this work, we showed that EAs can be used to generate GEs. We explored how viable this approach was for use in online generation regarding runtime. We presented a first FF to define complexity in GEs, which can be built upon to optimize player experience and understand complexity in future work. Our user study showed that our EA implementation allows authoring GEs of specific complexity. Tested within a city-building game, we created GEs that players perceived as significantly different in the required steps and interim products, and that impacted the odds of winning the game.

The results further suggest that rating complexity is difficult without direct comparison. Alternatively, complexity may require larger differences when embedded in a game (because participants of the online survey perceived the complexity difference outside of gameplay). This difficulty in perceiving differences in complexity within gameplay may facilitate using GEs for complexity adjustment to target difficulty or challenge—avoiding issues of too obvious game adjustments.

Our results further show that many players in this genre enjoy complex GEs, supporting literature on the connection between challenge and enjoyment. However, it also provides evidence for low complexity of GEs in games still coinciding with high enjoyment. Future work will have to explore whether this enjoyment is derived from the low complexity itself, or whether players find enjoyment in other challenging game mechanics apart from the GE.

In summary, this research provides a first proof-of-concept toward developing GEs via EAs with targeted complexity. The results support the possibility of reliably creating GEs of specific complexity to facilitate game development and user testing. Further, it lays the foundation for research on player-perceived GE complexity in games and informs our understanding of GE complexity.

Acknowledgment

The authors would like to thank all participants in the study and survey and our colleagues for helpful comments on early iterations of the fitness function.

References

[1] E. Adams, *Fundamentals of Game Design*. London, U.K.: Pearson Education, 2014.
[2] Wube Software, "Factorio," Game [macOS], Prague, Czech Republic, 2016, Wube Software, Prague, Czech Republic.
[3] A. Drachen, J. Riley, S. Baskin, and D. Klabjan, "Going out of business: Auction house behavior in the massively multi-player online game Glitch," *Entertainment Comput.*, vol. 5, no. 4, pp. 219–232, 2014.
[4] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *Proc. Int'l Conf. Foundations Digit. Games*, 2013, pp. 213–220.
[5] W. L. Raffe, F. Zambetta, and X. Li, "A survey of procedural terrain generation techniques using evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–8.
[6] S. M. Lucas and G. Kendall, "Evolutionary computation and games," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 10–18, Feb. 2006.
[7] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 172–186, Sep. 2011.
[8] J. Togelius, R. De Nardi, and S. M. Lucas, "Towards automatic personalised content creation for racing games," in *Proc. IEEE Comput. Intell. Games*, 2007, pp. 252–259.
[9] R. Hunicke, "The case for dynamic difficulty adjustment in games," in *Proc. Int. Conf. Adv. Comput. Entertainment Technol.*, 2005, pp. 429–433.
[10] E. Adams and J. Dormans, *Game Mechanics: Advanced Game Design*. San Francisco, CA, USA: New Riders, 2012.
[11] V. Lehdonvirta and E. Castronova, *Virtual Economies: Design and Analysis*. Cambridge, MA, USA: MIT Press, 2014.
[12] V. Lehdonvirta, "Virtual economics: Applying economics to the study of game worlds," in *Proc. Conf. Future Play*, Lansing, MI, USA, 2005.
[13] J. Dormans, "Simulating mechanics to study emergence in games," *Proc. Art. Intell. the Game Des. Process.*, vol. 2, no. 6.2, pp. 5–2, 2011.
[14] C. Stephens and C. Exton, "Measuring inflation within virtual economies using deep reinforcement learning," in *Proc. 13th Int. Conf. Agents Art. Intell.*, 2021, vol. 2, pp. 444–453.
[15] R. Hunicke and V. Chapman, "AI for dynamic difficulty adjustment in games, 5/2004," in *Proc. Assoc. Advance. Artif. Intell.*, 2004, pp. 2–5.
[16] S. Abuhamdeh and M. Csikszentmihalyi, "The importance of challenge for the enjoyment of intrinsically motivated, goal-directed activities," *Pers. Soc. Psychol. Bull.*, vol. 38, no. 3, pp. 317–330, 2012.
[17] P. Sweetser and P. Wyeth, "Gameflow: A model for evaluating player enjoyment in games," *Comput. Entertainment*, vol. 3, no. 3, p. 3, 2005.
[18] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun, "Emotion assessment from physiological signals for adaptation of game difficulty," *IEEE Trans. Syst. Man Cybern., Part A, Syst. Humans*, vol. 41, no. 6, pp. 1052–1063, Nov. 2011.
[19] R. Lopes and R. Bidarra, "Adaptivity challenges in games and simulations: A survey," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 2, pp. 85–99, Jun. 2011.
[20] G. Van Lankveld, P. Spronck, H. J. Van Den Herik, and M. Rauterberg, "Incongruity-based adaptive game balancing," in *Proc. Adv. Comput. Games*, Berlin, Germany, 2009, pp. 208–220.
[21] A. Denisova and P. Cairns, "Adaptation in digital games: The effect of challenge adjustment on player performance and experience," in *Proc. Annu. Symp. Comput.- Hum. Interaction Play*, 2015, pp. 97–101.
[22] C. Jennett et al., "Measuring and defining the experience of immersion in games," *Int. J. Hum. Comput. Stud.*, vol. 66, no. 9, pp. 641–661, 2008.
[23] C. Klimmt, T. Hartmann, and A. Frey, "Effectance and control as determinants of video game enjoyment," *Cyberpsychol. Behav.*, vol. 10, no. 6, pp. 845–848, 2007.
[24] T. J. Tijs, D. Brokken, and W. A. IJsselsteijn, "Dynamic game balancing by recognizing affect," in *Fun and Games*. Berlin/Heidelberg, Germany: Springer, 2008, pp. 88–93.
[25] N. Yee, "The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments," *Presence: Teleoperators Virtual Environments*, vol. 15, no. 3, pp. 309–329, 2006.

[26] M. Newheiser, "Playing fair: A look at competition in gaming," *Strange Horiz.*, 2009, Art. no. 1.
[27] K. Rogers et al., " KickAR: Exploring game balancing through boosts and handicaps in augmented reality table football," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, ser. CHI '18. New York, NY, USA, 2018, pp. 1–12.
[28] E. Andersen et al., "The impact of tutorials on games of varying complexity," in *Proc. SIGCHI Conf. Hum. Factors in Comput. Syst.*, ser. CHI '12. New York, NY, USA, 2012, pp. 59–68, doi: 10.1145/2207676.2207687.
[29] J. Smeddinck, K. M. Gerling, and S. Tiemkeo, "Visual complexity, player experience, performance and physical exertion in motion-based games for older adults," in *Proc. 15th Int. ACM SIGACCESS Conf. Comput. Accessibility*, ser. ASSETS '13. New York, NY, USA, 2013, doi: 10.1145/2513383.2517029.
[30] G. Aloupis, E. D. Demaine, A. Guo, and G. Viglietta, "Classic Nintendo games are (computationally) hard," *Theor. Comput. Sci.*, vol. 586, pp. 135–160, 2015.
[31] M. Lafond, "The complexity of speedrunning video games," in *Proc. 9th Int. Conf. Fun with Algorithms (FUN 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2018, pp. 1–19.
[32] N. D. Bowman, "Editorial: Video games as demanding technologies," *Media Commun.*, vol. 7, no. 4, pp. 144–148, 2019, doi: 10.17645/mac.v7i4.2684.
[33] A. E. Eiben et al., *Introduction to Evolutionary Computing*. Berlin/Heidelberg, Germany: Springer, 2003, vol. 53.
[34] C. H. Tan, K. C. Tan, and A. Tay, "Dynamic game difficulty scaling using adaptive behavior-based AI," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 4, pp. 289–301, Dec. 2011.
[35] M. J. Nelson, J. Togelius, C. Browne, and M. Cook, "Rules and mechanics," in *Procedural Content Generation in Games*. Berlin, Germany: Springer, 2016, pp. 99–121, doi: 10.1007/978-3-319-42716-4_6.
[36] T. Mahlmann, J. Togelius, and G. N. Yannakakis, "Towards procedural strategy game generation: Evolving complementary unit types," in *Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2011, pp. 93–102, doi: 10.1007/978-3-642-20525-5_10.
[37] U. Schöning, *Algorithmik*. Berlin, Germany: Springer, 2001.
[38] D. Makinson, *Sets, Logic and Maths for Computing*. Berlin/Heidelberg, Germany: Springer, 2012.
[39] K. A. De Jong, *Evol. Computation: A Unified Approach*. Cambridge, MA, USA: MIT Press, 2006.
[40] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.
[41] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.
[42] J. Nielsen, "Response times: The 3 important limits," 1993, added update from 2014. Accessed: 11 Apr. 2018. [Online]. Available: https://www.nngroup.com/articles/response-times-3-important-limits/
[43] J. Nielsen, *Usability Engineering*. San Mateo, CA, USA: Morgan Kaufmann, 1994.
[44] T. Adams, "Dwarf Fortress," Game [macOS], Silverdale, USA, Aug. 2006, Bay 12 Games, Silverdale, USA.
[45] Blue Byte Software, "The settlers online," Game [Browser], Montreuil, France, Sep. 2011, Ubisoft Entertainment SA, Montreuil, France.
[46] Maxis, "SimCity," Game [Windows], Redwood City, CA, USA, Mar. 2013, Electronic Arts, Redwood City, CA, USA.
[47] Colossal Order, "Cities: Skylines," Game [macOS], Stockholm, Sweden, Mar. 2015, Paradox Interactive, Stockholm, Sweden.
[48] S. Lundgren and S. Björk, "Neither playing nor gaming: Pottering in games," in *Proc. Int'l Conf. Foundations Digit. Games*. New York, NY, USA, 2012, pp. 113–120, doi: 10.1145/2282338.2282363.
[49] S. A. Alharthi, O. Alsaedi, Z. O. Toups, T. J. Tanenbaum, and J. Hammer, "Playing to wait: A taxonomy of idle games," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2018, pp. 1–15, doi: 10.1145/3173574.3174195.
[50] A. Denisova, P. Cairns, C. Guckelsberger, and D. Zendle, "Measuring perceived challenge in digital games: Development & validation of the challenge originating from recent gameplay interaction scale (CORGIS)," *Int. J. Hum. Comput. Stud.*, vol. 137, 2020, Art. no. 102383, doi: 10.1016/j.ijhcs.2019.102383.
[51] J. G. Gascon, S. M. Doherty, and D. Liu, "Investigation of videogame flow: Effects of expertise and challenge," *Proc. Hum Factors Ergon Soc Annu. Meet.*, Los Angeles, CA, USA, vol. 59, no. 1, pp. 1853–1857, 2015.
[52] W. Van den Hoogen, K. Poels, W. IJsselsteijn, and Y. de Kort, "Between challenge and defeat: Repeated player-death and game enjoyment," *Media Psychol.*, vol. 15, no. 4, pp. 443–459, 2012.
[53] ConcernedApe, "Stardew Valley," Game [OS X], London, England, 2016.