Evolving Interactive Narrative Worlds

Justus Robertson[1], John Heiden[1], Rogelio E. Cardona-Rivera[2]

[1] School of Interactive Games and Media, Rochester Institute of Technology, Rochester, NY, USA
[2] Division of Games, University of Utah, Salt Lake City, UT, USA
jjrigm@rit.edu, jdh5563@rit.edu, r.cardona.rivera@utah.edu

Abstract

An interactive narrative is bound by the context of the world where its story takes place. However, most work in interactive narrative generation takes its story world design and mechanics as given, which abdicates a large part of story generation to an external world designer. In this paper, we close the story world design gap with an evolutionary search framework for generating interactive narrative worlds and mechanics. Our framework finds story world designs that accommodate multiple distinct player roles. We evaluate our system with an action agreement ratio analysis that shows worlds generated by our framework provide a greater number of in-role action opportunities compared to story worlds randomly sampled from the generative space.

Introduction

There is a long history and broad diversity of approaches to automatically generating interactive stories for game environments, from sequenced story beats(Mateas and Stern 2002), to search(Nelson and Mateas 2005), automated planning(Riedl, Saretto, and Young 2003), optimization(Nelson et al. 2006), and many more. These systems are called *experience managers*(Riedl and Bulitko 2013) because they provide a targeted interactive experience for an end-user, who typically plays the role of a character within an interactive story. However, the properties and quality of stories these systems generate are influenced by the game worlds where the stories unfold. No matter how fast, elegant, or complex an experience manager is, its output is bound by the context and affordances provided by its story world design.

In *procedural content generation* (PCG) there is a much larger focus on the process of game level and mechanic generation, grounded in how these design choices influence gameplay possibilities. One popular approach to procedurally generating game levels and mechanics is *evolutionary search*(Togelius et al. 2011), which explores a generative space of design artifacts by optimizing an evaluation function that maps designs to gameplay outcomes. Additionally, there is work that mixes ideas from experience management and PCG. For example, the Planning Domain Definition Language (PDDL)(McDermott et al. 1998), a common representation for plan-based story worlds, was used to model automated game mechanic generation(Zook and Riedl 2014) and evolutionary search was used to find role-specific interactive plot graphs(Giannatos et al. 2011). However, no work both: 1.) generates novel interactive narrative world designs; and 2.) uses an evaluation function to find designs that maximize desirable gameplay outcomes.

In this paper, we bridge the gap between PCG and experience management with a software architecture that performs evolutionary search through a space of narrative world designs, specified in PDDL. Evolutionary search is guided by an evaluation function that maximizes the utility of distinct artificial player personas, similar to those presented by-Holmgård et al.(2015). We evaluate the world designs found with evolutionary search by comparing them to a second set of PDDL artifacts randomly sampled from the generator's generative space. Artifacts are compared in terms of an Action Agreement Ratio (AAR)(Holmgård et al. 2014) metric between different personas. We find that evolutionary search produces story world artifacts that better accommodate the different play styles when compared to a random sample, leading to lower agreement between personas in the evolutionary search worlds.

Related Work

Much work in the space of automated level and mechanic design is focused on action-based game genres, with a few outliers that merge PCG ideas with narrative generation. In this section, we start with an overview of traditional game level and mechanic PCG, then discuss work that has applied PCG techniques to narrative generation.

Level PCG Early work in the area of game level PCG applied evolutionary search to generating race car tracks that maximize entertainment value relative to a particular player(Togelius, De Nardi, and Lucas 2007). *Polymorph*, another early level PCG system, generated 2D platformer game levels at different difficulty levels based on a model of player skill(Jennings-Teats, Smith, and Wardrip-Fruin 2010). These are both examples of *experience-driven PCG*, where game content is generated based on its predicted effect on the player. These early themes of evolutionary search(Togelius et al. 2011) and experience-driven

PCG(Yannakakis and Togelius 2011) have come to be dominant themes within the level generation community. Our system fits within both of these approaches: we use evolutionary search to find interactive narrative game worlds that accommodate multiple playstyles human players can adopt.

Mechanic PCG Another common topic within the PCG community is generating mechanics that govern how game worlds behave. An early system in this area is *Game-o-Matic*(Treanor et al. 2012), which created playable game metaphors for a given concept map data structure, provided by a human author. Another early system is *Mechanic Miner*(Cook et al. 2013), which used reflection and evolutionary search to dynamically change mechanics for a platformer game based on online player interactions. Nielsen et al.(2015) also used evolutionary search to generate mechanics, defined in the Video Game Description Language (VGDL)(Ebner et al. 2013). More recent approaches include *Angelina*(Cook, Colton, and Gow 2016) and *Gemini*(Summerville et al. 2018). Both systems generated game mechanics based on a semantic understanding of the rules. Like several of these approaches, we use evolutionary search paired with a declarative language model to generate world mechanics. The main difference is: we generate worlds for interactive narratives instead of action-based games.

Narrative PCG Finally, there are examples of PCG systems that use concepts from the narrative generation community. Zook and Riedl(2014) generate high-level game mechanics, modeled in PDDL. The system does not have any narrative reasoning or application. It is included here because of its use of PDDL, which is a common modelling language in the plan-based narrative generation community. The more direct example of narrative PCG is Giannatos et al.(2011), a system that uses evolutionary search to evolve story graph data structures that maximize suspense. However, this approach is focused on evolving stories rather than the world and mechanics in which the story takes place. In this paper, we use evolutionary search to generate story worlds for interactive narratives to take place inside.

System Overview

Our goal is to search a space of game world designs to find instances that afford given playstyles. To accomplish this goal, we built a system that performs evolutionary search through a space of generated PDDL worlds, using Monte Carlo tree search (MCTS) personas to score world designs. This system consists of three components:

1. A PDDL(McDermott et al. 1998) world generator that combines 32 individual fragments together to produce a generative space of 1024 unique combinations.
2. A set of four MCTS-based player personas, similar to those presented inHolmgård et al.(2015): the Warrior, Collector, Sneaker, and Speedrunner.
3. An evolutionary search process that navigates the generative PDDL world space based on the utility scores of the four MCTS personas on the world designs.

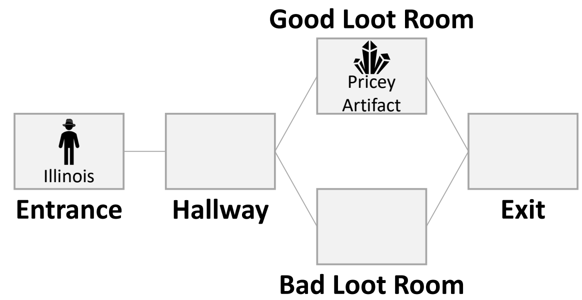

Figure 1: A diagram of the base world. There are five rooms: the Entrance, Hallway, Good Loot Room, Bad Loot Room, and Exit. The player, as Illinois, begins at the Entrance. The Pricey Artifact is located at the Good Loot Room. The player's goal is to grab the Pricey Artifact and reach the Exit.

We present the details of each system component before providing an Action Agreement Ratio (AAR)(Holmgård et al. 2014) analysis to evaluate the results.

Story World Generation

The first step of evolutionary search is to create a space of world design artifacts. In our case, these artifacts are game worlds, consisting of initial states and action mechanics, specified in PDDL. To create this pool of artifacts, we built a custom PDDL world generator with a generative space of 1024 artifacts. The generator is an additive process that combines 32 different hand-authored PDDL fragments together with a base world. The different possible combinations of the 32 fragments mixed with the base world results in the generative space of $32^2 = 1024$ full PDDL world variations.

Base World The base world is the simplest PDDL game world in the generative space. It is the template from which all other worlds are built:

1. There is one hero character controlled by the player, named Illinois Smith. Illinois can: (a) move between connected rooms; (b) pick up items; and (c) attack enemies.
2. There are five rooms: the Entrance, Hallway, Good Loot Room, Bad Loot Room, and Exit. The player begins at the Entrance. The Entrance is connected to the Hallway. The two loot rooms are connected in a diamond pattern between the Hallway and Exit.
3. There is a Pricey Artifact at the Good Loot Room.

The base world is pictured in Figure1. The player's objective is to grab the Pricey Artifact and reach the Exit. However, the base world is just one of the 1024 possible PDDL world designs the generator is capable of creating.

PDDL Fragments In addition to the base world, we defined 32 PDDL fragments. Rather than full playable worlds, these fragments contain isolated action and state definitions meant to be combined with the base world and each other to create new variations. These additions include: objects, initial state literals, predicates, and actions. Additionally, if any two actions defined in separate fragments have the same name and set of terms, the two actions are merged into a single definition by performing a union of the precondition and effect sets. The rule we followed when designing the fragments is: all information is strictly additive, so all possible combinations of PDDL fragments work together.

Fragments include: placing multiple enemies at different locations; a coffin that can be opened to find an artifact; a sand pit that can be dug into with a shovel to find an artifact; a bottomless pit that requires a whip to cross with, an enemy on the other side; a lever that opens a secret tunnel at the Entrance leading straight to the Good Loot Room; a dog that can be petted; the ability to be tired and take a nap; a portable cassette tape player that plays music; etc. A short description of all 32 fragments is given in the Appendix.

Many of the mechanics in the PDDL fragments interact with the player personas defined in the next section. Many others do not, but could interact with other possible personas. For example, the dog fragment will not change any of our persona utilities, but would increase the potential utility of a persona maximizing the number of animals petted before reaching the Exit with the Pricey Artifact.

Player Personas

Similar toHolmgård et al.(2015), we define four Monte Carlo tree search (MCTS) player personas to automatically playtest and score our worlds based on their persona-specific goals. Our four personas share the same overall MCTS search algorithm, but use different utility functions to score the results of their playtraces. All four personas share the base objectives of taking the Pricey Artifact and reaching the Exit. Additionally, each persona has at least one other individualized goal:

1. Warrior (W): is rewarded for each enemy it attacks. When an enemy is attacked, it becomes *vanquished*. The Warrior receives a reward relative to the number of vanquished enemies in the final state.
2. Collector (C): is rewarded for each artifact it collects. When an artifact is picked up, the character *has* the artifact. The Collector receives a reward relative to the number of artifacts they have in the final state.
3. Sneaker (S): is punished for being in a room with an enemy. The Sneaker is co-located with an enemy if they are *at* the same location. The Sneaker receives a punishment relative to the number of states in their playtrace where they are co-located with an enemy.
4. Speedrunner (R): is punished for each action it takes. This punishment is relative to the number of actions in their playtrace. The Speedrunner punishment can be combined with the other personas, to create a Warrior (WR), Collector (CR), and Sneaker (SR) that are incentivized to minimize their actions.

Together, these personas are used to score PDDL worlds during evolutionary search and perform the final AAR analysis. The baseline versions of the personas are used during evolutionary search and the speedrunner variants (WR, CR, and SR) are used during the AAR analysis. This is done to amplify the secondary objective reward signal during evolutionary search, then converge on the Speedrunner behavior in the absence of other objectives during the final analysis. This convergence will lead to high AAR values if individual goals are not satisfied in the game world.

Evolutionary Search

The final component is to run evolutionary search on our generative space to find PDDL world artifacts that support each of our four personas. For evolutionary search to be effective, each artifact needs a corresponding *genotype* at some level of *directness*, and *locality* must exist between the artifacts. A genotype is an abstract representation of generated artifacts that can be manipulated by evolutionary search. Directness is how closely the genotype represents its corresponding artifact. Locality means that changes to an individual gene correspond to equal and consistent changes to its artifact. Search also needs an evaluation function to score the different artifacts produced and search parameters to control the size of the population and offspring.

Representation Our narrative world genome is represented as a string with up to 32 unique characters. Each character represents a PDDL fragment and toggles the fragment's presence in the full PDDL artifact. If a character is present in a given genome string, the corresponding fragment is combined with the base world and all other present fragments to create the final PDDL artifact. If a character is not in the string, the corresponding fragment is ignored. In this way, all 1024 PDDL worlds can be represented with a string of up to 32 unique characters. This representation is mostly indirect, which reduces the search space complexity. It is similar to sequencing hand-crafted level "chunks" in traditional game level PCG. The representation has strong locality, as toggling an individual character on or off in the genome string represents adding or modifying a small number of action mechanics and initial state changes.

Evaluation We use a *simulation-based* evaluation to score worlds during search: the game worlds are played by artificial agents and ranked according to playtrace metrics. In our case, each of the four MCTS personas explore each world and then record the utility of their final playtrace. These four utilities are then combined into a final fitness function that represents how well the PDDL world accommodates persona goals. To ensure each persona contributes equally to the final fitness score, we track the highest and lowest score for each persona across all worlds encountered over all search sessions. We then normalize each persona's utility to a decimal value between 0 and 1 based on where the score falls in the persona's range of observed utility values. We multiply each normalized utility by 25 and add the values together, which gives a final fitness score in the range of 0 to 100, with each persona contributing between 0 and 25 points.

Search Parameters and Details The codebase is implented in C#. All tests took place on a PC laptop with a 12th Gen Intel Core i7-12700H processor, 32GB RAM, and a 930GB hard drive. Search sessions took place over a 14 day window. In this time, the PDDL fragments were tested and expanded. Data was collected for the score ranges used in the fitness function. The final search session was run for 96 hours with a μ and λ of 10, where μ is the size of the population kept between generations and λ is the size of the population generated through reproduction. This leads to a total population of 20 artifacts in each generation.

Each MCTS persona was allowed 1,000 search iterations for each generated world using the UCT formula and an exploration parameter of $\sqrt{2}$. Random rollouts ended whenever Illinois reached the Exit room. Final utilities were calculated according to the persona's evaluation function based on the playtrace and final state, then added to the MCTS "win" column during backpropagation. After 1,000 iterations, a final search was conducted with the MCTS exploration parameter set to 0. The utility of this final search was normalized and summed with the other personas, according to the search evaluation procedure.

Search Results

After 96 hours, evolutionary search completed 28 iterations, explored 274 unique worlds (26.76% of the generative space), and converged on a 100-score world at iteration 22. The average score of the first population was 34.35, while the average score of the 28th population was 91.80.

Before testing, we predicted the best genes would contain the characters (!, $, a, c, e, p, s, t) and not contain the characters (g, i, j, k, l, o, q, r, u, z). Figure2 shows a diagram of *!$acepst*, which is the world in this predicted optimal set with the fewest active genes. We found four genes with perfect scores. These genes all contain/exclude the predicted characters, along with variations of the neutral characters (with respect to our personas) *m* and *n*.

AAR Evaluation

In the final stage of this work, we perform an analysis to show the worlds we find through evolutionary search better afford the playstyles of our four personas when compared to a pool of worlds randomly sampled from the generative space. To perform this analysis, we use the Action Agreement Ratio (AAR)(Holmgård et al. 2014) metric. We calculate AAR on persona playtraces across the two sets of worlds, then perform a series of statistical tests. The statistical tests broadly support our hypothesis: that worlds found by evolutionary search will have a lower AAR compared to worlds randomly sampled from the generative space.

Action Agreement Ratio

AAR is a ratio that represents how often two personas choose the same action. AAR begins with a playtrace generated from some persona *A*. For every state in the playtrace, a second persona *B* is queried for the action it would perform in the given state. AAR is the final ratio between the number of times *A* and *B* choose the same action, divided by the total number of states in the playtrace. A score of 1.0 represents two personas that agree on every decision in a playtrace, while a score of 0.0 represents perfect disagreement. In this analysis, we will use the Speedrun personas (WR, CR, SR, and R) so that Illinois will default to completing the base objectives as quickly as possible in the absence of secondary persona objectives.

Hypothesis

If evolutionary search has correctly identified worlds that maximally accommodate secondary persona goals, then randomly sampled worlds will have less secondary goals for personas to complete compared to worlds found with evolutionary search. Without the presence of secondary goals, personas will default to completing the main objectives as quickly as possible. This will result in higher AAR agreement in worlds without secondary objectives compared to worlds that accommodate persona-specific goals. This leads to an empirical hypothesis:

General Hypothesis: *If evolutionary search has found worlds that accommodate each of our player personas, then the AAR between personas in the worlds found by evolutionary search will be lower than the AAR of worlds randomly sampled from the generative space.*

Data Collection

To collect the AAR data, two sets of game worlds were used: 1. worlds produced by evolutionary search and 2. worlds randomly sampled from the generative space. We used 10 worlds for each set. For each world, AAR values were collected by first running each of the Speedrun personas (WR, CR, SR, and R) for 5,000 search iterations using the UCT formula and an exploration parameter of $\sqrt{2}$. A full playtrace was then run using an exploration parameter of 0, which is the exemplar playtrace. Each state within the playtrace is given to the four personas, who have 1 second to perform exploratory search before running a final round with an exploration parameter of 0. The final result is returned as an action to take in the given state. If the action matches the one taken by the original persona in the exemplar playtrace, 1 is added to the AAR numerator. In either case, 1 is added to the AAR denominator for each playtrace action.

One important aspect of AAR is self-agreement between two runs of the same persona. Since our worlds are deterministic and the MCTS exploration parameter is set to 0, two playtraces of the same persona on the same world will yield identical action sequences until a state is reached that has not been added to the MCTS tree, triggering a random rollout. Once a random rollout begins, AAR will reflect how likely the same action can be chosen at random twice in a given state, which is based on the branching factor. However, if self-agreement is 1 then all actions reflect choices made by the persona based on MCTS information. For the AAR analysis to be accurate, it is important that the self-agreement of both world sets is comparable to reflect the same mix of MCTS signal vs. random rollout noise.

The average self-agreement across all personas in the 10 Evolutionary Search worlds is 99.42%. This number is high because the worlds found by evolutionary search have an average 9.2 out of 32 gene positions active, which are worlds that are easier for MCTS to exhaustively search. When randomly sampling worlds for the second set, it is important to sustain a self-agreement similar to the Evolutionary Search

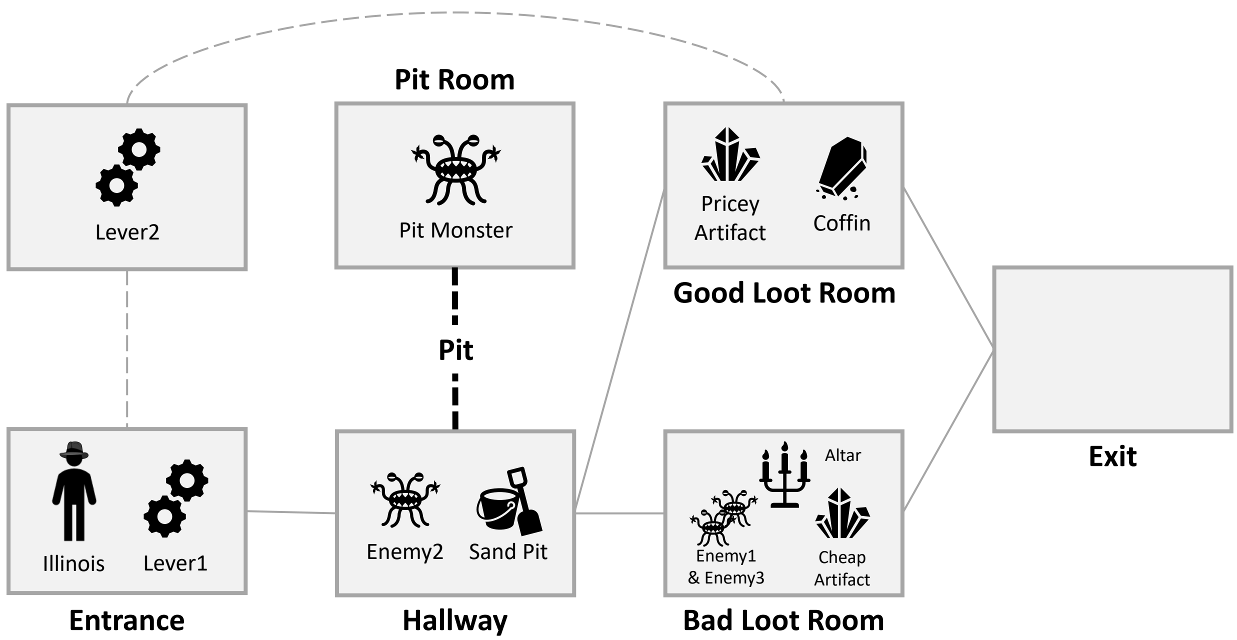

Figure 2: A diagram of world *!$acepst*. It contains 4 different enemies for the Warrior to vanquish, 5 different artifacts for the Collector to find, a path with no enemies for the Sneaker to navigate, and a minimum sequence of 4 actions for the Speedrunner.

AAR in Evolutionary Search Worlds	**WR**	**CR**	**SR**	**R**
Warrior Speedrun (WR)	1.000	0.328	0.403	0.373
Collector Speedrun (CR)	0.261	1.000	0.284	0.306
Sneaker Speedrun (SR)	0.452	0.484	1.000	0.661
Speedrunner (R)	0.186	0.372	0.558	0.977

Table 1: AAR between all personas (WR, CR, SR, and R) within worlds found with Evolutionary Search. Shows worlds found via evolutionary search trend toward lower than coin-flip AARs (i.e., 0.500) between *different* personas, with the exception of the SR and R personas.

AAR in Randomly Sampled Worlds	**WR**	**CR**	**SR**	**R**
Warrior Speedrun (WR)	1.000	0.500	0.577	0.462
Collector Speedrun (CR)	0.521	1.000	0.589	0.479
Sneaker Speedrun (SR)	0.564	0.577	0.974	0.564
Speedrunner (R)	0.375	0.475	0.525	1.000

Table 2: AAR between all personas (WR, CR, SR, and R) Randomly Sampled from the generative space. Unlike the trend shown by Table1, this table shows most personas agree with each other at a rate higher than coin-flip odds (i.e., 0.500) in Randomly Sampled worlds.

set. To this end, we rejected any world that caused average self-agreement across the Random Sample worlds to be further than 1% away from that of the Evolutionary Search set. This selection criteria biases the random sample towards worlds with a similar state complexity as the evolutionary search set, but ensures an apples-to-apples MCTS signal to rollout noise ratio. Using this process, the final average Random Sample self-agreement value is 99.35%, which is 0.07% less than the Evolutionary Search set.

AAR data for worlds found with Evolutionary Search is summarized in Table1. The data for Randomly Sampled worlds is summarized in Table2. The full AAR dataset is provided in the Appendix.

Data Analysis

As mentioned, our overall hypothesis is: the average AAR between personas in worlds found by evolutionary search will be *lower* than worlds randomly sampled from the generative space. The initial data reported in Tables 1 and 2 confirm this trend: most personas agree at a rate lower than coin-flip odds in the Evolutionary Search worlds and higher than coin-flip odds in Randomly Sampled worlds. Our in-depth evaluation tests whether these AAR proportions are statistically-significant. We conduct two analyses for this evaluation: one of the stratified (i.e., per persona) effect of evolutionary search on persona-driven gameplay and another on the cumulative effect.

Stratified Analysis Our stratified analysis compares the AARs across Tables 1 and 2 on a per-persona basis. A given *reference* persona is analyzed against all other *comparison* personas (excluding self-comparison) by counting the number of times the reference and the comparison agreed on an action over the number of actions taken. This stratified AAR is computed twice: once for the worlds found via evolutionary search, and once for randomly sampled worlds. Thus, a given persona $X \in \{\mathrm{WR}, \mathrm{CR}, \mathrm{SR}, \mathrm{R}\}$ yields two AARs: one AAR p_1^X that represents its behavior as effected by the evolutionary search, and another AAR p_2^X representing its

Persona		Action Agreement Ratio (AAR)		Two-Sample Test for Equality of Proportions		Effect Size Analysis	
Reference	Comparison	Evo Search: p_1	Rand Sampling: p_2	χ^2 (df) using H_A	p-value [95% CI]	$p_1 - p_2$	Cohen's h
WR	CR, SR, R	$0.368 = 148/402$	$0.513 = 120/234$	$12.695(1)$	$< .001^*[-1.000, -0.078]$	-0.145	0.2924807
CR	WR, SR, R	$0.284 = 114/402$	$0.530 = 116/219$	$36.819(1)$	$< .001^*[-1.000, -0.179]$	-0.246	0.5070318
SR	WR, CR, R	$0.532 = 161/248$	$0.577 = 135/234$	$0.83788(1)$	$.18[-1.000, 0.036]$	-0.045	0.08989863
R	WR, CR, SR	$0.372 = 48/129$	$0.458 = 55/129$	$1.9063(1)$	$.08369[-1.000, 0.016]$	-0.086	0.1752595
Cumulative		$0.3655 = 409/1119$	$0.528 = 426/807$	$50.338(1)$	$< .001^*[-1.000, -0.125]$	-0.1625	0.3281354

* Significant under Bonferroni correction for multiple comparisons ($\alpha = 0.05$; $\alpha_{\text{Bonferroni}} = 0.05/5 = 0.01$).

Table 3: Data analysis results. The first four rows summarize the stratified analysis and the last row cumulative analysis.

behavior as effected by the randomly sampled worlds.

These two AARs are themselves compared using the Two-Sample Test for Equality of Proportions(Cohen 1988). For the four personas this means evaluating four individual hypotheses, each of the following form:

$$\mathrm{H}_0 : p_1^X = p_2^X \qquad \text{for} X \in \{\mathrm{WR}, \mathrm{CR}, \mathrm{SR}, \mathrm{R}\}$$
$$\mathrm{H}_A : p_1^X < p_2^X \quad \text{using} \alpha = 0.05 (\text{w/ Bonferroni correction})$$

To assess the magnitude of the difference between the proportions, we relied on Cohen's h, itself defined as the difference between the angular transformation of the proportions(Cohen 1988):

$$h^X = \varphi_1^X - \varphi_2^X$$
$$\text{where:} \varphi_i^X = 2 \arcsin(\sqrt{p_i^X})$$

Cumulative Analysis This is the aggregate of the stratified analysis, and follows the same operationalized form of the null/alternate hypotheses described earlier.

Results

All five operationalized hypotheses are evaluated with the R statistical software package, version 4.3.0 (Already Tomorrow). The results are presented in Table3. Cumulatively, we found support for our hypothesis: the average AAR of Personas in worlds found by Evolutionary Search (i.e., p_1) is lower than the average AAR in Randomly Sampled worlds (i.e., p_2). The magnitude of this difference is small (Cohen's $h^{\mathrm{WR}} \approx 0.33$) but statistically significant.

When stratified per persona, we see the largest effect on the Collector Speedrun (CR) Persona (Cohen's $h^{\mathrm{CR}} \approx 0.51$, medium difference), followed by the Warrior Speedrun (WR) Persona (Cohen's $h^{\mathrm{WR}} \approx 0.29$, small difference); neither the Sneaker Speedrun (SR) nor baseline Speedrun (R) personas exhibited statistically-significant differences in their behavior within worlds found by evolutionary search relative to worlds found via random sampling. This suggests that the strength of the cumulative effect is due in large part to the WR and CR personas.

Discussion

We find cumulative support for our general hypothesis and statistically-significant differences for the stratified WR and CR personas. We don't observe a difference for the SR and R personas, but this is expected given the above coin-flip (0.558 and 0.661) AAR ratios between the SR and R personas shown in Table1. The ratios between these two personas are high because there is not much difference between them, even when both of their individual goals are accounted for. The Speedrun persona tries to grab the Pricey Artifact and reach the Exit in as few turns as possible. The Sneaker-Speedrun persona tries to do the same thing, while also avoiding enemies. This often leads to these two personas having a small AAR delta, no matter the world.

Consider world *!$acepst*, shown in Figure2. The Speedrunner will move to the Hallway, where there is an enemy. This initial move will differ from the Sneaker, who wants to avoid the enemy. However, the Speedrunner will not enter another room with an enemy after this first move to the Hallway, so the two personas will agree on actions for the rest of the playtrace. Alternatively, the Sneaker will use Lever1 to enter the Secret Passage to the Good Loot Room to bypass Enemy2 at the Hallway. After the first two actions of opening and entering the Secret Passage, it will be more expensive for the Speedrunner to go back to its original route than to follow the Sneaker's path forward. Once the two personas converge on a path of action, they have perfect agreement to the end of the playtrace. And this series of events is what happens for world *!$acepst* in our dataset: the Speedrunner to Sneaker AAR is $3/4$ and the Sneaker to Speedrunner AAR for *!$acepst* is $4/6$.

This similarity between the Sneaker and Speedrunner personas leads to the larger than coin-flip SR/R values in both Table1 and Table2. It also explains why the stratified SR and R analyses are not statistically significant: these high AAR values between SR and R are not a product of evolutionary search, but a feature of all worlds across the generative space. Additionally, the SR/R to WR/CR values in the same rows are noticeably lower in the Evolutionary Search table (0.452, 0.484, 0.186, 0.372) when compared to the Randomly Sampled table (0.564, 0.577, 0.375, 0.475). This indicates that a difference exists between both the SR and R persona when compared to the WR and CR personas, but the uniform similarity between the SR and R personas throughout the generative space causes non-statistically-significant SR and R persona outcomes.

Overall, we find support for our general hypothesis. The design of randomly sampled words "coerce" personas into taking actions similar to others, so these worlds do not afford the expression of unique play-styles. However, worlds found by evolutionary search afford the simulation-based expression of different gameplay types. This demonstrates that careful construction of evolutionary-guided search can yield meaningfully different design spaces—i.e., those that support persona expression as measured by diverging AAR.

Limitations and Future Work

We have shown that evolutionary search is capable of finding procedurally generated worlds that accommodate a range of different playstyles, as implemented with MCTS personas. There are a number of ways this work can be used moving forward, including: 1.) expanding the generative space of the system to search through a larger collection of PDDL worlds, and 2.) creating mixed initiative authoring tools for human-machine co-creation. In this section, we discuss each of these possible future directions.

Expanding Generative Space

One drawback of the current project is the size of the generated worlds and the scope of the generator. While 1024 possible worlds is strong for an initial system, it would be nice to search through a larger portion of the PDDL possibility space. There are both smaller, easier changes that can be made to the current indirect generator as well as larger modifications that will make the representation more direct.

Fragment Generator The current PDDL generator can be modified to add arbitrarily many high-level features, like rooms, connections, enemies, etc. Some gene positions could control what features are active, like the current fragments, while other positions handle the number of objects and how they are distributed across the map. This would drastically widen the space of generative worlds without changing the current PDDL fragment design.

Full PDDL Generation The size of generative space corresponds to how *direct* the genome representation is. Our current representation and the above modification are relatively indirect, meaning the genome representation is abstracted away from the actual features of the final artifacts. Our artifacts are text-based PDDL documents, so the most direct representation would be the individual ASCII characters and their location in the artifact document. However, this would lead to a generative space of all text documents, where many are not valid PDDL. A representation one step more indirect is where all artifacts are valid PDDL but the system can add or remove any kind of object, type, predicate, action, precondition, effect, fluent, etc. that it wishes, so the generative space is the set of all PDDL documents.

One problem with this approach is the issue of semantics. All combinations of worlds in our current generative space have an external narrative and game context of a galactic treasure hunt. The objects, actions, and world configurations in our PDDL fragments are hand-engineered to resemble common tropes in the adventure and science fiction narrative genres, along with common mechanics in text-based and point-and-click adventure games. However, the vast majority of artifacts in the space of all PDDL documents do not have this ludo-narrative information baked in.

A middle ground between our fragment approach and navigating the full space of PDDL documents would be a system capable of generating novel ludo-narrative PDDL objects to populate the fragment pool. This approach is similar to recent work in the interactive narrative planning community, which has automatically generated narrative actions(Porteous et al. 2015) and object types(Porteous et al. 2020) using linguistic information from WordNet(Christiane 1998). These techniques could be used to automatically generate novel PDDL objects with a consistent narrative theme to constrain a larger generative space to narrative-oriented PDDL documents.

Authoring Tools

Extensions to our system could allow human designers to benefit from MCTS personas and evolutionary search, and vice-versa. These extensions could include a state visualizer/editor. Although humans can directly edit PDDL files, like we have, it would be easier to have a state visualizer. Given a PDDL world, the visualizer could make a diagram of the world layout and mechanics. This is similar to work in the interactive narrative planning community that automatically visualizes game states based on an underlying PDDL state representation(Robertson and Young 2015). The interface could also allow human authors to reconfigure the design of the initial state and actions to create new world designs. If the editor is capable of translating editor designs back into PDDL documents, the document could be used as the starting point for new evolutionary search sessions.

Conclusion

In this paper, we showed that evolutionary search can find interactive narrative worlds that uniquely accommodate player personas. We created a world generator that combines 32 different PDDL fragments into 1024 possible worlds. We introduced four different MCTS personas that optimize different goals and use these personas to guide evolutionary search. Finally, we ran an Action Agreement Ratio analysis to show that worlds found through evolutionary search have lower AARs compared to worlds randomly sampled from the generative space. This shows that the worlds found with evolutionary search better accommodate the playstyles represented by MCTS personas compared to worlds sampled from the generator.

PDDL Fragments

There are a total of 32 PDDL fragments that can be combined with each other and the base world for a total of 1024 different world designs. Each fragment is associated with a character that is added or removed from the genome, which is represented by a string of enabled fragments. This section provides a detailed account of each fragment, listed by its corresponding gene character:

! Places an artifact called the Cheap Artifact at the Bad Loot Room. This gene gives the Collector persona an additional artifact to collect, which raises its utility.

Evolutionary Search Data											
Persona	!$acemnpst	!$acenpst	!$acepst	!$acempst	!$acenps	!$abcenps	!$acemnps	!$acempqst	!$acehkmpst	!$acekpst	Total
WW	14 / 14	14 / 14	12 / 12	12 / 12	13 / 13	14 / 14	13 / 13	14 / 14	14 / 14	14 / 14	134 / 134
WC	3 / 14	5 / 14	3 / 12	2 / 12	4 / 13	4 / 14	6 / 13	9 / 14	6 / 14	2 / 14	44 / 134
WS	6 / 14	5 / 14	3 / 12	5 / 12	4 / 13	7 / 14	7 / 13	7 / 14	5 / 14	5 / 14	54 / 134
WR	5 / 14	5 / 14	5 / 12	5 / 12	8 / 13	6 / 14	5 / 13	4 / 14	3 / 14	4 / 14	50 / 134
CW	2 / 12	2 / 12	2 / 12	4 / 12	3 / 13	5 / 14	4 / 13	4 / 14	5 / 17	4 / 15	35 / 134
CC	12 / 12	12 / 12	12 / 12	12 / 12	13 / 13	14 / 14	13 / 13	14 / 14	17 / 17	15 / 15	134 / 134
CS	2 / 12	2 / 12	2 / 12	3 / 12	3 / 13	4 / 14	5 / 13	4 / 14	7 / 17	6 / 15	38 / 134
CR	4 / 12	4 / 12	3 / 12	4 / 12	4 / 13	3 / 14	4 / 13	3 / 14	8 / 17	4 / 15	41 / 134
SW	3 / 7	3 / 7	3 / 6	2 / 6	2 / 5	4 / 6	2 / 5	2 / 6	4 / 7	3 / 7	28 / 62
SC	3 / 7	4 / 7	2 / 6	4 / 6	2 / 5	2 / 6	2 / 5	4 / 6	3 / 7	4 / 7	30 / 62
SS	7 / 7	7 / 7	6 / 6	6 / 6	5 / 5	6 / 6	5 / 5	6 / 6	7 / 7	7 / 7	62 / 62
SR	4 / 7	4 / 7	4 / 6	4 / 6	4 / 5	4 / 6	4 / 5	4 / 6	3 / 7	6 / 7	41 / 62
RW	0 / 4	0 / 4	1 / 4	2 / 4	2 / 4	0 / 4	1 / 4	1 / 5	1 / 5	0 / 5	8 / 43
RC	3 / 4	2 / 4	1 / 4	2 / 4	2 / 4	1 / 4	1 / 4	2 / 5	1 / 5	1 / 5	16 / 43
RS	1 / 4	2 / 4	3 / 4	2 / 4	3 / 4	3 / 4	3 / 4	3 / 5	2 / 5	2 / 5	24 / 43
RR	3 / 4	4 / 4	4 / 4	4 / 4	4 / 4	4 / 4	4 / 4	5 / 5	5 / 5	5 / 5	42 / 43

Random Sample Data											
Persona	23ltuv	4giknp	4bfsuvz	ehopu	23bdfghjlpsy	12lortz	!bipqsv	dmntv	afgnrs	1fmnr	Total
WW	6 / 6	10 / 10	7 / 7	8 / 8	8 / 8	8 / 8	10 / 10	7 / 7	7 / 7	7 / 7	78 / 78
WC	3 / 6	5 / 10	2 / 7	3 / 8	1 / 8	8 / 8	5 / 10	5 / 7	3 / 7	4 / 7	39 / 78
WS	6 / 6	3 / 10	4 / 7	5 / 8	1 / 8	6 / 8	4 / 10	7 / 7	5 / 7	4 / 7	45 / 78
WR	2 / 6	3 / 10	4 / 7	3 / 8	2 / 8	5 / 8	4 / 10	4 / 7	4 / 7	5 / 7	36 / 78
CW	2 / 6	2 / 7	3 / 6	1 / 3	2 / 6	8 / 8	4 / 11	4 / 7	6 / 12	6 / 7	38 / 73
CC	6 / 6	7 / 7	6 / 6	3 / 3	6 / 6	8 / 8	11 / 11	7 / 7	12 / 12	7 / 7	73 / 73
CS	2 / 6	4 / 7	2 / 6	3 / 3	1 / 6	7 / 8	4 / 11	6 / 7	7 / 12	7 / 7	43 / 73
CR	3 / 6	3 / 7	3 / 6	3 / 3	1 / 6	6 / 8	4 / 11	4 / 7	3 / 12	5 / 7	35 / 73
SW	6 / 6	3 / 14	5 / 7	1 / 3	2 / 5	5 / 6	7 / 16	7 / 7	4 / 7	6 / 7	46 / 78
SC	5 / 6	5 / 14	4 / 7	3 / 3	1 / 5	5 / 6	4 / 16	6 / 7	5 / 7	7 / 7	45 / 78
SS	6 / 6	13 / 14	7 / 7	3 / 3	5 / 5	6 / 6	15 / 16	7 / 7	7 / 7	7 / 7	76 / 78
SR	4 / 6	6 / 14	5 / 7	3 / 3	3 / 5	3 / 6	7 / 16	5 / 7	3 / 7	5 / 7	44 / 78
RW	2 / 3	1 / 5	0 / 3	1 / 3	1 / 3	2 / 4	0 / 5	3 / 4	3 / 5	2 / 5	15 / 40
RC	2 / 3	2 / 5	1 / 3	3 / 3	1 / 3	2 / 4	1 / 5	3 / 4	1 / 5	3 / 5	19 / 40
RS	1 / 3	2 / 5	1 / 3	3 / 3	2 / 3	2 / 4	1 / 5	3 / 4	3 / 5	3 / 5	21 / 40
RR	3 / 3	5 / 5	3 / 3	3 / 3	3 / 3	4 / 4	5 / 5	4 / 4	5 / 5	5 / 5	40 / 40

$ Places an enemy named Enemy1 at the Bad Loot Room. This gene gives the Warrior an enemy to attack, which raises its utility. It also gives the Sneaker a room to avoid.

1 Creates a new room, called the Hidden Loot Room. Like the other loot rooms, the Hidden Loot Room is connected in a diamond pattern between the Hallway and Exit.

2 Creates a new room, called the Loop Room. The Loop Room is connected to the Entrance and the Hallway, creating a circuit between these initial three rooms.

3 Creates a new room, called the DK Room. The DK Room is only connected to the Entrance. This room is named after *Donkey Kong Country*, which allows players to backtrack from the start of levels to find collectables.

4 Creates a new room, called the Secret Room. The Secret Room is connected to the Good Loot Room. This is an additional alcove to hide collectables.

a Places an Altar object at the Bad Loot Room and allows Illinois to perform the pray action. When Illinois prays in a room with an altar, a hidden Holy Artifact is revealed. This gene gives the Collector persona another artifact to collect.

b Places a beverage, called Slurm, in Illinois' inventory and introduces the thirsty status effect. A held beverage can be consumed to remove the thirsty status effect.

c Places a closed coffin at the Good Loot Room. A Coffin Artifact is inside the closed coffin. The coffin can be opened, which allows access to the Coffin Artifact. This gives the Collector persona another artifact to collect.

d Creates a dog, named Spot, at the Hallway. Illinois can pet an animal if they are in the same room.

e Adds two additional enemies: Enemy2 is added to the Hallway and Enemy3 to the Bad Loot Room. This gives the Warrior persona two extra enemies to attack, but makes it impossible for the Sneaker persona to reach the exit without encountering an enemy (unless the 't' gene is active).

f Illinois is holding a sandwich and is hungry. Hunger can be alleviated by eating food.

g Adds a force field that prevents Illinois from taking artifacts. The force field is applied to the Cheap Artifact, created by gene **'!'**. This prevents the Cheap Artifact from being collected by Illinois, which negatively impacts the Collector's utility by removing an artifact to find.

h Illinois is wearing a fedora. A fedora is a famous hat worn by explorers named after US states...

i Sets the position of Enemy1 (also used by gene '$') to all locations in the base world. This projects the same enemy into multiple locations at once, which negatively impacts the Sneaker's utility. However, the Warrior still only has one enemy to vanquish so this gene does not benefit the Warrior.

j Places Enemy1 (added to the Bad Loot Room by gene **'$'** and other positions by 'i') in a cell. This prevents the enemy from being attacked by the Warrior, while still negatively impacting the Sneaker's score.

k Illinois is given a key that must be used to unlock passageways leading to the Exit. This negatively impacts the Speedrunner persona by introducing an additional action that must be performed to move from the Good Loot Room to the Exit after picking up the Pricey Artifact.

l Rooms are now unlit, like a cave. In order to take an item in a room, the room must be lit. Only the Entrance begins lit. Illinois has a lighter object that will light a room when used. This negatively impacts the Speedrunner persona by introducing an additional action that must be taken in the Good Loot Room before the Pricey Artifact can be taken.

m Creates a salesman at the Entrance. This feature can be combined with other genes to offer Illinois world-specific items for purchase.

n Illinois is tired. The tired status effect can be removed by taking the nap action, which has no other effect.

o Creates three enemy phantoms: one each at the Hallway, Bad Loot Room, and Good Loot Room. The phantoms are enemies to be avoided by the Sneaker but cannot be attacked by the Warrior. This negatively impacts the Sneaker (unless the 't' gene is present) and does not benefit the Warrior.

p Creates a room called the Pit Room. The Pit Room is connected to the Hallway by a special connection called a Pit. Illinois is given a Whip which is used to swing across the Pit to reach the Pit Room. There is an enemy located at the Pit Room, named Pit Monster. This gives the Warrior an extra enemy to attack, which raises its utility.

q The Hallway is now filled with Quicksand. If Illinois enters a room with Quicksand, they become trapped. To escape, they must use a Whip in their inventory.

r The Hallway is now a trap. To leave the Hallway, Illinois must use a Rock in their inventory to hit a button in the Hallway to disable the room's trap.

s A sandpit is added to the Hallway and Illinois is given a shovel. If Illinois digs in the sandpit, they will find the Sandy Artifact. This is another artifact for the Collector to find.

t Creates a secret passage that connects the Entrance to the Good Loot Room. To access the passage, two levers must be pulled. The first lever is at the Entrance and opens the pathway to the Secret Passage room. The second lever is in the Secret Passage and opens the pathway to the Good Loot Room. This allows the Sneaker persona to avoid any enemies located at the Hallway (for example, those added by 'e' and 'o'). In conjunction with 'e', this gene allows the Warrior to benefit from the extra enemies while allowing the Sneaker an alternate route around them.

u Adds an artifact to the Hallway and requires artifacts have the *stable* status effect to be picked up. All artifacts begin unstable. In order for any artifacts to be stabilized and collected, the 'v' gene must be present.

v Gives Illinois a gadget that stabilizes an artifact. This gadget can be used when the **'u'** gene is present to stabilize artifacts so they can be collected. However, the gadget only has a single charge. So even 'u' and 'v' together will result in the Collector finding less artifacts.

w Illinois begins with a Walkman: a portable cassette player, famously used by space explorers from Earth...

x Illinois begins with a xylophone. The xylophone starts with 'x' and is used to play music.

y Illinois has yarn than can be unspooled to mark locations as they explore the ruins, similar to the Greek myth of Theseus and the Minotaur.

z A magic charm is given to Illinois that turns all the enemies nice. This prevents enemies from being attacked but they still exist, so this negatively impacts both the Warrior and the Sneaker's scores.

AAR Dataset

The tables in this section show the playtrace data used in the AAR analysis. There are two tables of data: worlds produced by evolutionary search and worlds selected by our sampling process. The far-left column of each table contains two characters that correspond to the two personas being compared. The remaining columns are for each of the worlds in the set, labeled by their genome. Each entry is an AAR ratio: the numerator is the number of matching actions and the denominator is the total number of actions in the playtrace. The far-right column is the sum of all ratios in the row.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. #2303650. We also wish to thank the anonymous reviewers who were tremendously helpful with their comments during peer review.

References

Christiane, F. 1998. WordNet: An Electronic Lexical Database. *Computational Linguistics*, 292–296.

Cohen, J. 1988. *Statistical power analysis for the behavioral sciences*. Hillsdale, New Jersey, USA: Erlbaum.

Cook, M.; Colton, S.; and Gow, J. 2016. The ANGELINA Videogame Design System — Part I. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(2): 192–203.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Applications of Evolutionary Computation: 16th European Conference, EvoApplications 2013, Vienna, Austria, April 3-5, 2013. Proceedings 16*, 284–293. Springer.

Ebner, M.; Levine, J.; Lucas, S. M.; Schaul, T.; Thompson, T.; and Togelius, J. 2013. Towards a Video Game Description Language. *Dagstuhl Follow-Ups*, 6.

Giannatos, S.; Nelson, M.; Cheong, Y.-G.; and Yannakakis, G. 2011. Suggesting New Plot Elements for an Interactive Story. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 25–30.

Holmgård, C.; Liapis, A.; Togelius, J.; and Yannakakis, G. 2015. Monte-Carlo Tree Search for Persona Based Player Modeling. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 8–14.

Holmgård, C.; Liapis, A.; Togelius, J.; and Yannakakis, G. N. 2014. Evolving Personas for Player Decision Modeling. In *IEEE Conference on Computational Intelligence and Games*, 1–8. IEEE.

Jennings-Teats, M.; Smith, G.; and Wardrip-Fruin, N. 2010. Polymorph: A Model for Dynamic Level Generation. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6, 138–143.

Mateas, M.; and Stern, A. 2002. A Behavior Language for Story-Based Believable Agents. *IEEE Intelligent Systems*, 17(4): 39–47.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL - The Planning Domain Definition Language. Technical Report CVC TR98003/DCSTR1165, Yale Center for Computational Vision and Control, New Haven, CT.

Nelson, M.; and Mateas, M. 2005. Search-Based Drama Management in the Interactive Fiction Anchorhead. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 1, 99–104.

Nelson, M. J.; Roberts, D. L.; Isbell Jr., C. L.; and Mateas, M. 2006. Reinforcement Learning for Declarative Optimization-Based Drama Management. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 775–782.

Nielsen, T. S.; Barros, G. A.; Togelius, J.; and Nelson, M. J. 2015. Towards generating arcade game rules with VGDL. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, 185–192. IEEE.

Porteous, J.; Ferreira, J. F.; Lindsay, A.; and Cavazza, M. 2020. Extending Narrative Planning Domains with Linguistic Resources. In *International Conference on Autonomous Agents and Multiagent Systems*, 1081–1089.

Porteous, J.; Lindsay, A.; Read, J.; Truran, M.; and Cavazza, M. 2015. Automated Extension of Narrative Planning Domains with Antonymic Operators. In *International Conference on Autonomous Agents and Multiagent Systems*, 1547–1555.

Riedl, M.; and Bulitko, V. 2013. Interactive Narrative: An Intelligent Systems Approach. *AI Magazine*, 34(1): 67–77.

Riedl, M.; Saretto, C. J.; and Young, R. M. 2003. Managing Interaction Between Users and Agents in a Multi-Agent Storytelling Environment. In *International Conference on Autonomous Agents and Multiagent Systems*, 741–748.

Robertson, J.; and Young, R. M. 2015. Automated Gameplay Generation from Declarative World Representations. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 11, 72–78.

Summerville, A.; Martens, C.; Samuel, B.; Osborn, J.; Wardrip-Fruin, N.; and Mateas, M. 2018. Gemini: Bidirectional generation and analysis of games via asp. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, 123–129.

Togelius, J.; De Nardi, R.; and Lucas, S. M. 2007. Towards Automatic Personalised Content Creation for Racing Games. In *2007 IEEE Symposium on Computational Intelligence and Games*, 252–259. IEEE.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3): 172–186.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-Matic: Generating Videogames that Represent Ideas. In *Workshop on Procedural Content Generation in Games*, 1–8.

Yannakakis, G. N.; and Togelius, J. 2011. Experience-Driven Procedural Content Generation. *IEEE Transactions on Affective Computing*, 2(3): 147–161.

Zook, A.; and Riedl, M. O. 2014. Automatic Game Design via Mechanic Generation. In *AAAI Conference on Artificial Intelligence*.