Adaptive Generation of Characters for Tabletop Role Playing Games

Fabrício Guedes Faria, Leonardo Tórtoro Pereira, Cláudio Fabiano Motta Toledo
Instituto de Ciências Matemáticas e de Computação (ICMC)
University of São Paulo
São Carlos, Brazil
fabricio.guedes.faria@usp.br, leonardop@usp.br, claudio@icmc.usp.br

***Abstract*—This paper tries to solve the problem where automated character creation for Tabletop RPGs is completely random and does not fully pleases certain players, especially those that have difficulty in being creative and adapting characters to their personal tastes. To accomplish that, the usage of a Evolutionary Algorithm (EA) was proposed, in conjunction with the user's input (positive or negative) during generations, to create characters focused on having traits that are catered to each individual's taste. The generated traits are "Race", "Class", "Size", "Visual Trait", "Personality" and "Quirk". The *fitness* calculation for each individual was made using trait tables with weights previously defined by the programmer, and posteriorly by the user's evaluation of the characters during execution.**

The project was made using the Unity game engine with C# language, to take advantage of the tool's easy interface creation. It was produced as a phone application, for ease of distribution and general use.

The application was tested with a variety of users, and the results, presented in the form of graphs, indicate that the rejection rate was almost null, with the great majority of users recognizing the gradual improvement of suggestions between generations, and evaluating positively the characters generated by the algorithm.

***Keywords*-evolutionary algorithm; procedural content generation; character generation; tabletop RPG; user input;**

I. Introduction

Games are an ancient activity, and have been constantly evolving over the ages, adapting to the technology available at each point in time to allow more complex and engaging activities [1][2]. In 1980, one type of game, *Wargames*, which existed in some form since the 1780s, saw a spike in popularity, led primarily by the brand of miniature gaming *Warhammer* [3].

They served as main inspiration to the creation of Gary Gygax's *Dungeons & Dragons*, the first Tabletop Role Playing Game (TTRPG). Since then, various systems and settings were created to accommodate different personal tastes, but the biggest is still *D&D*, which in its 5th edition has over 14.8 million players worldwide [4][5].

TTRPG's are multiplayer pen-and-paper games based on imagination. Players take on the role of a fictional character in a fantasy world, and undertake quests in search of loot, glory, and fun. One player takes the role of the Dungeon Master, controlling the monsters, guiding the plot and calling for dice rolls to determine the outcome of player's decisions [6].

These games can help develop soft skills in its players, specially the strengthening of bonds, respect, acceptance and socialization, as observed in Jaques and Bisol study in a brazilian high-school [7]. Cook et al. used TTRPGs to engage students into active literacy practices and increase the students' collaboration between themselves [8]. Rivers et al. also observed that TTRPG players have higher levels of empathy [9]. A summary of the many human skills which can be learned through RPG is presented by Daniau, and includes creativity, spontaneity, exploration of identity, the search for creative and novel solutions, communication, conflict resolution, diplomacy, etc. [10].

TTRPGs were also used to aid psychotherapeutic treatment, such as one case of a depressive, suicide schizoid personality of a 19 year old male reported by Blackmon [11]. Kim et al. used RPGs as a narrative therapy group counseling tool to reduce negative emotions and increase happiness in elementary school students [12].

Before the game's sessions, players must create their characters following a process defined by the system. It normally involves choosing numerical values for stats and skills, and also thinking of appearance and personality traits. Sometimes, players have a hard time in avoiding *cliches* and common tropes, and can resort to online character generators [13][14][15]. The main downside of said generators is that they only have the capacity to randomize characteristics, and do not take the user's personal tastes into account.

This work's objective is to use an Evolutionary Algorithm (EA) to generate interesting characters that adapts to the user's personal profile based on their positive or negative feedback between generations.

II. Related works

In order to adapt characters to the player's interests, we use the technique of Procedural Content Generation (PCG). That is, the creation of content using algorithms. This technique was first used in the 80's, with random seeds to define parameters for instantiating certain objects in a game. Nowadays, many famous games in the video-game industry use PCG to generate different contents, such as animations, textures and levels [16].

PCG is also an ever-growing field of scientific research, where many different computer science algorithms are used, covering areas such as neural networks, EAs, formal grammars, machine learning and others. These algorithms are used to automatically generate levels, artwork, sounds and music, mechanics, gameplay, rules, narratives, behavior of non-playable characters and simulation of player behavior. Some good examples of these algorithms and applications in the most recent literature can be found in the references we will detail next.

The work of Yannakakis et al. presents a good panorama of artificial and computational intelligence in games, while Risi and Togelius present a review of neuroevolution algorithms and Summerville et al. present some of the most recent works of PCG using machine learning [17][18][19]. Kybartas and Bidarra present "a survey on story generation techniques and authoring computation narratives", focusing on PCG of narratives and, lastly, Shaker et al. present an overview of PCG in games in their book, introducing some of the most common algorithms and applications on scientific and commercial games [20][16].

Few PCG applications were found that focused in the generation of characters of any kind. As most were focused in video-games, they approach the generation of Non-Player Characters (NPCs) or narratives involving them. In the master's thesis of Griffith, the author generates (or uses a random algorithm to generate) many different NPCs' personalities by adding different trait modules (i.e.: generous, violent, etc.) to each. They also are assigned wants and needs by the author, that are a vector of weights and a signed value indicating the amount of a resource the NPC wants or needs. Then, the author also defines a world state with workplaces and who they belong to, what actions and interactions were performed by actors, etc. Then, the whole system is simulated using the previously defined information, and the results of the actions of each NPC is represented and executed as a statechart. Then, the resulting state of the world after the interactions can be observed and the results show that the NPCs and their actions were considered consistent with their characters and traits by most users [21].

The work of Eccles, another master thesis, implements a believable system for the behavior of NPCs. Said model accounts for agents changing with experience and awareness as they interact with others and getting tired after working, for example. The emotional state of NPCs can be expressed through and action hierarchy i.e.: they can socialize in a nice or unpleasant way. Their personality is also displayed through behavior: consideration weights and state modifiers are generated procedurally and changes the agents preferences to specific actions from a broadly universal set of possible actions. Although said actions are predictable from the agent's personality, the predictability's level can be adjusted through pseudorandom number generation [22].

Franco et al. also propose a model for character generation, this time being a part of an interactive storytelling model [23]. The character model relies on the moral and order alignment, which, by *D&D* rules, affect their behaviour, and also on other common attributes in *D&D* such as strength and charisma. They also have attributes which influence the selection of goals and plans, such as sense of duty, spiritual commitment, affinity to take risks, and others. The attributes are evaluated by a similarity function that takes into account the attributes, goals and selected plans for each character and results in how suitable an action or goal is for a given character.

However, these models are focused in generating characters for digital games, not TTRPGs. For the specific context of TTRPGs, we found even fewer examples using computational-aided approaches to enhance their gameplay and fun. The works of Martin et al. [24] and Padovani et al. [25] being the most relevant ones found. The first uses reinforcement learning agents, using a Deep Q Network, to learn the rules of TTRPGs and play them. The agent model is composed of two models: a genre expectation model, trained using a corpus of stories from a genre related to the fictional world the agent will be placed in and providing the probability, according to the corpus, that certain events will happen after other ones; and a commonsense rules model, that prunes out candidate events not suitable for the current game state, and allows the agent to plan how current actions might affect the world state in the future. After training, the agent is able to convert a natural language sentence into an event and use it to update the agent's belief about the world. Then, the agent takes its turn, selects a new event using the learner and update the belief again. Then, the event is converted back into natural language for the human player [24].

The second work proposes an intelligent system able to select background music for TTRPG games. The Bardo system uses speech-recognition to transform the conversations between players in a session into text. This text is handled to a supervised learning algorithm to classify it into an emotion and select a song representing that emotion. The authors show that a simple Naive Bayes classifier has an accuracy good enough for the task and an experiment with users resulted with them preferring the system over the original musics chosen for a campaign displayed online [25].

No work involving the creation of characters for the TTRPGs context was found, neither for tools that help players create them, or help the Dungeon Master in creating NPCs for the story. The examples found were websites that generate random characters or personalities, all of them based on random seeds. The website *RanGen* offers different random generators for helping people build stories, including character's appearance, archetypes, family, personality, etc. [15]. The *NPC Generator* website offers a more TTRPG-specific NPC generator, allowing the user to choose a specific race, sex, alignment, plot hook and occupation for a

character or let it also be random, and then generate a whole text-based description for the NPC, a list of personality traits, ability attributes, relationship status and alignment tendencies, all of them at random, but considering the user's input if there is any [13]. The *RPG Tinker* website uses a similar approach, allowing users to choose some characteristics for a character or letting them all be random. But the information created is more centered in the *D&D 5e* format [14].

Observing the lack of user-centered character generators, especially in the domain of TTRPGs, we developed our character generation system, taking into account the most important character traits seen in TTRPGs settings, and observed in the previously mentioned generators, and also some of the attributes seen in the works of Griffith, Eccles and Franco et al. [21][22][23]. The major contribution of the system is that it is able to adapt its character generation to the user's liking and their personal view on what traits are best suited together. The details of the system's implementation and how it adapts to each user are described in the following section.

III. Model Overview

A "Character" in this paper is made up of 6 traits, these being "Class", "Personality", "Quirk", "Race", "Size" and "Physical Trait". Each one represents an important aspect of a player's character.

"Class" is the role a character has in the game. For example, a Barbarian is a mighty close-quarters warrior, while a Cleric's job is to heal allies and smite enemies with holy energy. "Personality" is a general description of how a character normally acts in most situations, i.e.: in a Cruel or Courageous way. "Quirk" describes one specific behaviour that makes this character unique, for example if they hate animals or thinks they are inside of a game or simulation. "Race" in the context of fantasy games means different types of sentient creatures with specific physical characteristics that exist in the world, such as Elves, Goblins or Humans. "Size" is the simplest trait, and defines how is the character's build, including height, body fat and muscle presence. "Physical Trait" represents one difference on this character's body that makes them different than other people but does not affect the game majorly (i.e.: one blind eye, hand burns or face scars).

Each trait can take one of many possible values, defined initially by us and unique for every trait. As an example, there are 11 possible Classes and 60 different Personalities. The generator selects one value for each trait and assigns it to a generated character. Some of these traits were inspired by the work of [21].

During generation, we need to consider the users' opinion in order to guide the selection of traits to ones that are of their best interest. For that purpose, we must trace a profile for the users before they actually see any final character. That is accomplished by having them answer a short questionnaire when first accessing the app, and saving it to a user profile. During next uses, the same profile may be selected to skip the questionnaire. An example of the profile selection screen is shown in Fig. 1.

Figure 1. Profile selection screen.

There are a total of 6 questions used to define the users' profile, designed to influence all of the generated traits. The users answer each question with a certain type of character they like in mind, for example how aggressive they are and what their favorite hobby is. Each question can be answered with one integer value between 1 and 5 (even for questions with descriptive answers), and they are saved to the users' profile for later use. The profile questions are shown in Table I, and an example with the app interface is shown in Fig. 2.

TABLE I. USER PROFILE QUESTIONNAIRE

1. How does your character deal with conflicts? (1-Talking, 5-With fists)				
1	2	3	4	5
2. What does your character believe in most?				
Magic	Science	Nature	In theirself	Religion
3. What is your character's favorite hobby?				
Sports	Books	Traveling	Combat	Gambling
4. What does your character thinks about rules? (1- Does not care about them, 5- Follows diligently)				
1	2	3	4	5
5. How frequently does your character like to interact with others? (1- Never, 5- Whenever possible)				
1	2	3	4	5
6. How good was your character's childhood? (1- Very hard, 5- Grew up very well)				
1	2	3	4	5

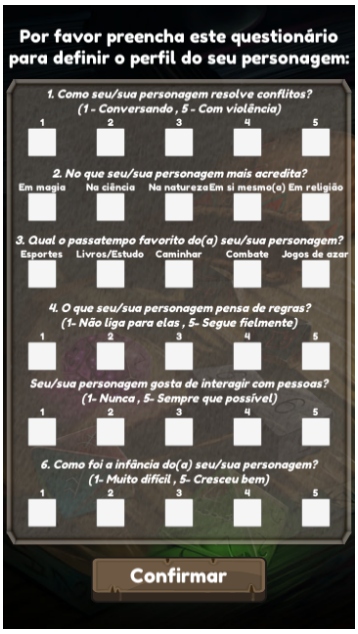

Figure 2. Profile questionnaire screen.

A. *Defining character traits*

Our generator follows the model of a genetic algorithm, which uses the concepts of genes and evolution of populations to reach the best possible outcome for the user. The chromosome's representation is shown in Fig. 3. It is an array of integers, where each position (or gene) is directly linked to a possible value for one trait of the generated character. To each one of these, a value is assigned out of all the possible ones for its trait. Every population will always have 100 individuals.

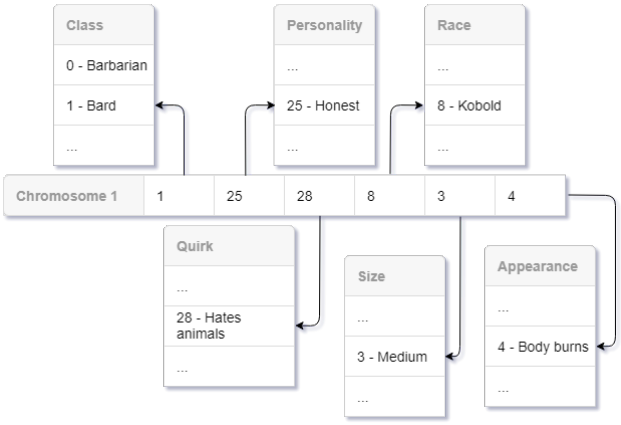

Figure 3. Representation of a chromosome.

In the first population, every individual receives random values for its genes. The algorithm calculates a fitness value for each chromosome, and sorts them based on this value. From that point on, whenever a new population is created, the algorithm selects two chromosomes from the current population, using a Roulette Selection, to act as parents for every new chromosome. Each pair of parents generates exactly one new chromosome. This chromosome's genes are defined using a Uniform Crossover technique, where each gene has an equal chance of being the same as that of parent 1 or parent 2. There also is a 10% chance of a mutation occurring for every gene: its value is changed to another random value.

The individuals created this way form the next population, and the old population is eliminated completely. No elitist selection strategy is applied because, since the best individual is shown to the user at every evaluation step and then shown again at the final selection step, we wanted to avoid making them see the same character repeatedly.

The population size and mutation value were chosen empirically as they generated the best results in previous experiments.

B. *Fitness calculation*

After a character is created and its traits defined, it needs to be evaluated on how well it conforms to the user's personal tastes. To do that, we use a fitness function that accesses weight tables based on the user's questionnaire answers and calculates a final number, which aims to be the highest possible.

We have one weight table for each combination of profile question and character trait (in total 36). On each cell, there is an integer value between 1 and 10 representing how likely this trait takes on the corresponding value when the user gives the corresponding answer to a profile question. The initial values are always between 1 and 10, but further values are not limited to this range. As exemplified in Table II, when the user answers the question "How aggressive is your character?" with a value of 5, the most likely Class to be suggested is "Barbarian", with a value of 10. It is important to note that other classes with lower values have some chance of being assigned to a final character, just not as likely.

The initial values of the table were defined by targeting common stereotypes for each different trait option. For example, the *Barbarian*'s initial *Aggression* value is 10, as that is the common trope of a barbarian character in RPGs. The user then alters those values based on their own biases, either maintaining tropes or breaking the norm. This process is further explained in section C.

The fitness of a given individual is calculated by a function, shown in Algorithm 1, that accesses all tables and adds the weight values corresponding to its generated values. The final sum represents how close that character is to the users' tastes: the higher the fitness value, the better.

This value is returned as the function result, and all generated characters are ordered by their fitness, and the

TABLE II. EXAMPLE OF A WEIGHT TABLE (CLASS\AGGRESSION).

Class\Aggression	1	2	3	4	5
Barbarian	1	2	5	8	10
Bard	2	4	6	4	2
Warlock	1	3	5	8	6
Ranger	1	2	8	6	3
Cleric	10	7	6	3	2
Druid	7	10	8	5	4
Sorcerer	2	4	8	7	6
Warrior	1	4	7	8	6
Rogue	4	6	7	6	5
Wizard	7	8	6	4	2
Paladin	4	8	7	6	5

best ones have a higher probability of being used to create the next population.

Algorithm 1 Algorithm for calculating the fitness of an individual.

procedure FITNESS(index, userAnswer)
Require: Index of a chromosome in the population ($index$) and vector of user answers ($userAnswer$)
  $genes \leftarrow chromo[index]$
  $fitness \leftarrow 0$
  repeat
    $table \leftarrow$ retrieve next weight table
    $k \leftarrow userAnswer[i]$
    $value \leftarrow table[genes[i]][k]$
    $fitness \leftarrow fitness + value$
  until all indexes of $genes$ have been accessed
  $Return fitness$
end procedure

C. User influence on the algorithm

When using the mobile application containing our character generator, the users are shown the traits of the best character from 10 different populations. The first character is always from the first generation, and subsequent characters are chosen after the 3 next generations. For each character, the users must answer if they think that the generated traits were consistent or not with their answers to the profile questionnaire. In order to do so they must use two buttons on the lower part of the screen, as shown in Fig. 4.

This input is used to update all weight tables and affects all further generated characters. Each cell with the corresponding trait value and profile answer has its weight value increased or decreased by 2 (value chosen empirically) depending on the users' input. An example is shown on Table III.

After 9 different characters have been shown to the user, and after every update to the weight tables have been applied, the generator shows a final character that should be the most recommended for this user. But the user may also have liked any of the previously shown characters, so we offer a way of them cycling through them all. This final screen is shown in Fig. 5, where there is also a button in the lower part of the screen that leads to the evaluation questionnaire.

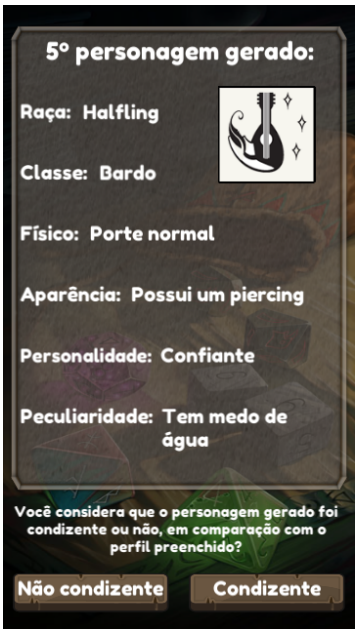

Figure 4. Example of a generated character and user input buttons. This character is a Halfling Bard, of normal size, that wears a piercing. They're confident, but afraid of water.

TABLE III. EXAMPLE OF A WEIGHT TABLE POSITIVE UPDATE.

Class\Aggression	1	2	3	4	5
Barbarian	1	2	5	8	10
Bard	2	4	6	4	2
...	.	.	.	.	.

Class\Aggression	1	2	3	4	5
Barbarian	1	2	5	8	12
Bard	2	4	6	4	2
...	.	.	.	.	.

IV. EVALUATION AND RESULTS

To evaluate the character generator, we uploaded the application in the Itch.io platform and disclosed the download link, which was made public, in social media platforms. A total of 28 people tried the application and answered a Google Forms survey asking questions about their knowledge of TTRPGs and their opinion about the application. As the experiment was conducted in Brazil, the app was primarily developed in Portuguese as well as the questionnaire, and their contents will be translated in this paper.

Fig. 6 shows how many users already played a TTRPG or, if they did not play, if they would be interested to. More than 70% of the respondents had already played TTRPGs, while for the ones that did not play, more than half (17,9% of the total) had the desire to play someday. So we can

Figure 5. Final app screen, showing the best generated character and directing to the evaluation questionnaire. This character is a Halfling Druid, that is muscly and mute. They're paranoid and bites their lips constantly.

consider the majority of the answers came from people with domain in the area or at least people interested in the topic. For the 20 people that played, they were asked the frequency at which they played, and the results are shown in Fig. 7. We can observe the frequency is very heterogeneous, with almost half of them playing at least once a week, and almost half playing with no regularity. This means that the answers come from an heterogeneous group of players, which is good for the randomness of the sample.

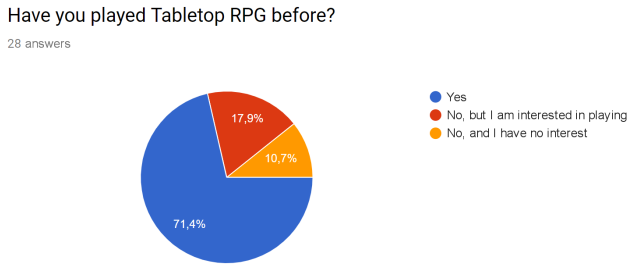

Figure 6. Answers for the question: Have you ever played TTRPGs?

Next, users were asked their opinion about the initial questionnaire that defined their profile before creating the characters. The answer was a 6-item Likert scale ranging from 0 (very bad) to 5 (very good) and is presented in Fig. 8. The results were very positive, with more than 96% of the respondents giving a positive value in the Likert scale (between 3 and 5). Only one gave a negative value, and it was only a little negative, being the value 2. Next, we asked users how much the created character matched the

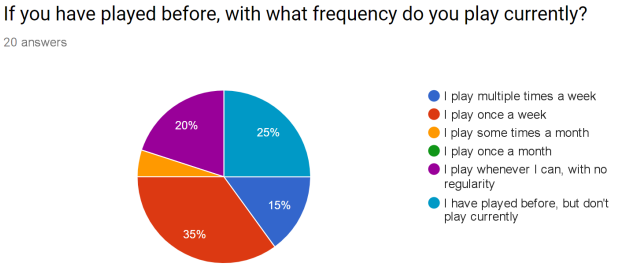

Figure 7. Answers for the question: With what frequency do you play TTRPGs?

profile they were envisioning when answering our profile-creation questionnaire. This also used a 6-item Likert scale, ranging from 0 (Does not match at all) to 5 (Matches totally). The results were more neutral, with the majority of answers (more than 85%) being positive, with only 3 being very little negative (value 2) and 1 moderately negative (value 1), as can be seen in Fig. 9.

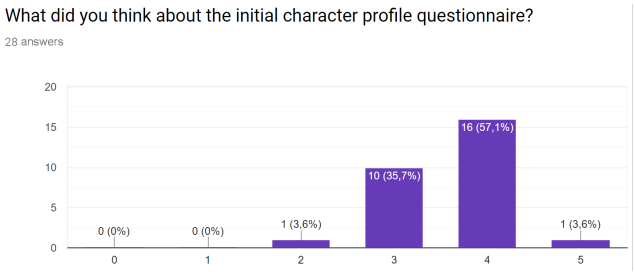

Figure 8. 6-item Likert scale opinion about the quality of the profile-creation questionnaire.

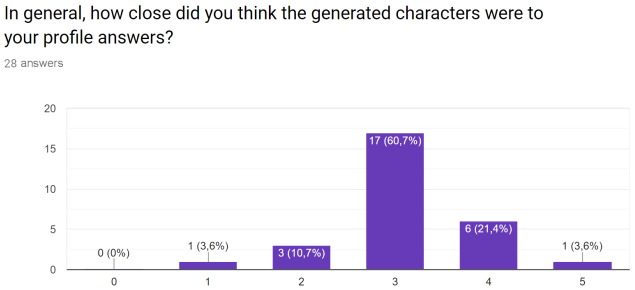

Figure 9. 6-item Likert scale opinion about how much the generated character matched the profile desired by the user.

The fifth question asked if the user perceived that the generated characters became progressively more likely to match their expected profile. With this, we are able to measure how much the EA's adaptation to each user was perceived and the answers are present in Fig. 10. As can be seen, more than 78% of the users perceived that the character creation was matching more their expected profile, meaning the EA's adaptation was, indeed, perceived and occured as expected.

Lastly, the users were asked how much they liked the generated characters, not considering if they matched or not the profile they had in mind for them. That is, if the character profile *per se* was likeable. The 6-item Likert scale ranged from 0 (did not like it) to 5 (liked it very much) and the answers are presented in Fig. 11. It can be seen that every answer was positive, with almost half of them being the most positive value, 5, meaning the users liked the character's very much.

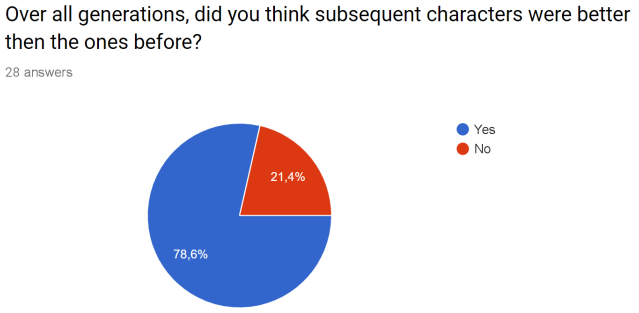

Figure 10. Users' opinion if they perceived that the generated characters' got closer to their expected profile at each new EA's generation.

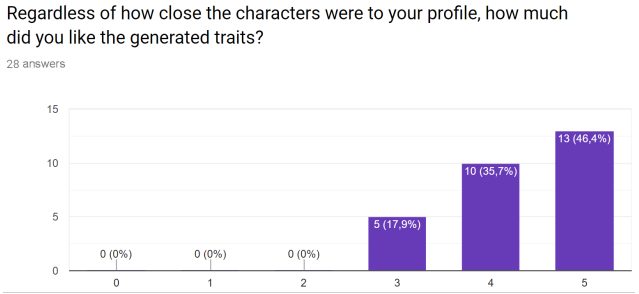

Figure 11. 6-item Likert scale opinion about how much the users liked the generated character not considering how well they matched the desired profile.

V. Conclusion

In this paper we show that the usage of an EA paired with user feedback to generate diverse and user-centered characters for TTRPGs sessions produces good results. Our approach gives RPG players a new tool to use when creating new characters for their games, which is able to adapt the characters to the users' views of what traits are better suited among themselves, and offers some variability while still maintaining consistency with their own tastes. This generator aims to help new players have an easier time beginning their TTRPGs sessions, while also helping veterans being more creative with their stories, by creating characters that correspond to their likes and dislikes and have their traits arranged as closely as what each user has in mind as the best, by collecting their input and adapting to them individually. We hope that by using it, together with other systems like Bardo, more people will be interested in playing TTRPGs and also benefit from the positive results that playing these games bring [25].

As further research, we intend to perfect the algorithm, try out different parameters and add more traits for the characters. We also plan to reformulate the questionnaire to have questions that are more straightforward and transparent do the user. Furthermore we want to reach a larger user base during later testing, to verify our product with more certainty.

References

[1] J. Huizinga, *Homo ludens a study of the play-element in culture*. Angelico Press, 2016.

[2] R. Guins and H. Lowood, *Debugging game history: a critical lexicon*. The MIT Press, 2016.

[3] G. COSTIKYAN, "The revolution began with paper," Apr 2006. [Online]. Available: http://www.escapistmagazine.com/articles/view/video-games/issues/issue_42/253-The-Revolution-Began-With-Paper.2

[4] Roll20, "The orr group industry report - q1 2018," Jun 2018. [Online]. Available: https://blog.roll20.net/post/174833007355/the-orr-group-industry-report-q1-2018

[5] Amazon, "Best sellers in fantasy gaming." [Online]. Available: https://www.amazon.com/Best-Sellers-Books-Fantasy-Gaming/zgbs/books/16211/ref=zg_bsnr_tab_t_bs

[6] Roll20, "Roll20 wiki," 2019. [Online]. Available: https://wiki.roll20.net/Introduction_To_Tabletop_RPGs

[7] R. R. Jaques and C. A. Bisol, "Role-playing games for common well-being in high school—notes from a study developed in southern brazil," *International Journal of Information and Education Technology*, vol. 9, no. 7, 2019.

[8] M. P. Cook, M. Gremo, and R. Morgan, "We're just playing: The influence of a modified tabletop role-playing game on ela students' in-class reading," *Simulation & Gaming*, vol. 48, no. 2, pp. 199–218, 2017.

[9] A. Rivers, I. E. Wickramasekera, R. J. Pekala, and J. A. Rivers, "Empathic features and absorption in fantasy role-playing," *American Journal of Clinical Hypnosis*, vol. 58, no. 3, pp. 286–294, 2016.

[10] S. Daniau, "The transformative potential of role-playing games—: From play skills to human skills," *Simulation & Gaming*, vol. 47, no. 4, pp. 423–444, 2016. [Online]. Available: https://doi.org/10.1177/1046878116650765

[11] W. Blackmon, "Dungeons and dragons: The use of a fantasy game in the psychotherapeutic treatment of a young adult," *American journal of psychotherapy*, vol. 48, pp. 624–32, 02 1994.

[12] KimBo-Kyung, BaeSung-Hoon, ChungEun-Jung, and HwangSoon-Taeg, "The effects of narrative therapy group counseling program using role-playing game format on happiness and negative emotions in children," *Journal of the Korea Convergence Society*, vol. 9, no. 9, pp. 271–284, 09 2018.

[13] Etiens, "Npc generator," acessed: 2019-07-08. [Online]. Available: http://www.npcgenerator.com/

[14] B. Freitas, "D&d 5e npc generator," acessed: 2019-07-08. [Online]. Available: https://rpgtinker.com/

[15] Ameronis, "Character generators," acessed: 2019-07-08. [Online]. Available: http://www.rangen.co.uk/chars/chargens.php

[16] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural content generation in games*. Springer, 2016.

[17] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 4, pp. 317–335, Dec 2015.

[18] S. Risi and J. Togelius, "Neuroevolution in games: State of the art and open challenges," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 1, pp. 25–41, March 2017.

[19] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. Hoover, A. Isaksen, A. Nealen, and J. Togelius, "Procedural content generation via machine learning (pcgml)," *IEEE Transactions on Games*, vol. PP, 02 2017.

[20] B. Kybartas and R. Bidarra, "A survey on story generation techniques for authoring computational narratives," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 239–253, Sep. 2017.

[21] I. Griffith, "Procedural narrative generation through emotionally interesting non-player characters," Master's thesis, Linnaeus University, 2018. [Online]. Available: http://lnu.diva-portal.org/smash/record.jsf?pid=diva2\%3A1231584\&dswid=-5564

[22] V. Eccles, "Procedurally generated background characters," Master's thesis, University of Dublin, 2017. [Online]. Available: https://scss.tcd.ie/publications/theses/diss/2017/TCD-SCSS-DISSERTATION-2017-036.pdf

[23] A. O. Franco, J. G. Maia, J. A. Neto, and F. A. Gomes, "An interactive storytelling model for non-player characters on electronic rpgs," in *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE, 2015, pp. 52–60.

[24] L. J. Martin, S. Sood, and M. Riedl, "Dungeons and dqns: Toward reinforcement learning agents that play tabletop role-playing games." in *INTWICED@ AIIDE*, 2018.

[25] R. Padovani, L. Ferreira, and L. Lelis, "Bardo: Emotion-based music recommendation for tabletop role-playing games," in *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2017. [Online]. Available: https://www.aaai.org/ocs/index.php/AIIDE/AIIDE17/paper/view/15909/15201