Q-Learning in Management-Tycoon Games: AI Rivals in *Ghost Writer*

Vincent Le
B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY, USA, 14623
vdl9926@rit.edu

Abstract

Tycoon games often have rivals to challenge the player, but most of them fall flat in their execution of creating a competent and compelling AI rival. Using a modified form of Q-Learning, *Ghost Writer* explores how to define a "learned" agent, compares its superior ability over a typical, random-based agent, and examines ways to tune learned agents using their tailored parameters.

1 Overview

Ghost Writer is a writing-based management-tycoon game in which the player competes to become a best-seller, stimulating the industry with new ideas while keeping themselves financially afloat. However, to get there, players will need to outperform their rivals—the AI (artificial intelligence) agents. In *Ghost Writer*, written works receive a score (and, consequently, sales) based on multiple factors: the compatibilities of their chosen topics, genres, and audience; how many points they allocated into each "focus" area during development (which compares to their chosen genre); how much total time the player dedicated to their work (which may vary based on the type of work: short story versus a novel, for instance); and the amount of generated errors left by the time of publishing. As the player progresses, the AI will progress with them at faster or slower rates using a modified form of Q-Learning. AI Agents will learn the intricacies of creating a written work that will produce the most sales in parallel to the player, provided the player can keep up.

2 What is *Ghost Writer?*

Ghost Writer is a 2D, management-tycoon game where the player takes up the mantle as an author within an undead world, researching topics and genres and testing their will as a writer as they make their way to become a bestseller. When making a written work, players will choose up to three topics and a single genre, (the numbers of maximum chosen topics and genres can be expanded later through research). They will also choose an audience (Children, Teens, Young Adults, Adults) and a work type (poem, short story, novel, screenplay, etc.). After choosing a concept, the player will enter stages of development where they will use sliders—ranging from values one-to-ten—to allocate points into specific categories of development (worldbuilding, dialogue, symbolism, etc..). The player will generate errors during the development cycle, unless nullified by traits selected when creating their avatar. After the development cycle, these errors can be amended through polishing, which happens automatically after the development cycle. Once all errors have been removed, continuing to polish will generate additional points for the final score. When the player chooses to release their work, they will be rated based on the compatibility of their topics, genres, and audience; how well their allocated points in their respective focus areas reflect on the given genre; the player's mastery of the topic and genre (which can be increased by using the same topic/genre multiple times); and how many errors they had when they released the work. After receiving their score, the player will gain money based on weekly sales. Their goal is to stay financially afloat while making the best work possible to outperform the competition—other AI agents.

3 Hypothesis

Management-tycoon games foster creativity, but most sacrifice challenges in order to do so. The AI rivals are often static and underwhelming—a side thought—which is disappointing considering that most Tycoon games denominate themselves as "simulations." By establishing more competitive AI rivals, players will be forced to think further into the gameplay rather than that provided in their own, isolated, world space; learned AI agents within a management-tycoon game will nurture a more competitive environment for the player.

4 Approach

4.1 What is Q-Learning?

To make AI agents incite more competition, they need to

provide a challenge. In terms of *Ghost Writer*, this means that the AI agent needs to learn how to play the game: learn the many combinations that will get them a good score. Q-Learning enables storing and updating information as actions are explored..

Q-Learning, as Ian Millington describes it in *AI for Games*, "treats the game world as a state machine, [...] the state should encode all the relevant details about the character's environment and internal data." (Millington 2019). The states within *Ghost Writer* describe stages of development, as well as subsets of information necessary. These stages are: Concept, Focus Area One, Focus Area Two, and Focus Area Three; sub-information resides in the Focus Area stages, in which states might also describe the specific genre chosen within the Concept phase. However, while typical Q-Learning has more constant iteration as states move between each other, because of *Ghost Writer*'s necessity for stages of development, this is unrealistic and nonoptimal. The phases of iterations of Q-Learning are compartmentalized to their specific stages of development, which are then chained together, meaning that states themselves already direct from one to another. That, coupled with the reality that the data within each state is only relevant to that state specifically (the only data passed through each state consistently is the Genre chosen during the Concept Phase), comparisons of new states as part of the Q-Learning equation is unnecessary. This changes the so-called "experience tuple" (Millington 2019):

$$Q\,(s,a) \;=\; (1-\alpha)Q(s,\,a) \;+\; \alpha(r \;+\; \gamma max(Q(s\text{'},a\text{'})))$$

Instead of using a new state as part of the algorithm, *Ghost Writer* uses the max Q value as reference, transforming the experience tuple into this:

$$Q\,(s,a) \;=\; (1-\alpha)Q(s,\,a) \;+\; \alpha(r \;+\; \gamma max(Q(s,\text{a})))$$

Consequently, the change in the experience tuple also transforms the "discount" parameter, which previously "[controlled] how much the Q-value of the current state and action depends on the Q-value of the state it leads to" (Millington 2019). Instead, now it discounts states based on the current highest-valued state seen, meaning that high discount values will attract states towards the max Q-value state, and low discount values will allow for more opportunity to expand away from the max Q-value state.

4.2 Q-Learning Implementation

For Q-Learning, *Ghost Writer* expands upon the implementation given by Millington in *AI for Games* (Millington 2019). Millington describes the implementation of a Q-Learning function that uses four tailored parameters: the learning rate *alpha*, the discount rate *gamma*, the randomness for exploration *rho*, and the length of the walk *nu* (Millington, 2019); each of these variables are within the range of [0-1]. However, as *Ghost Writer* separates the development of an entity's works through four, isolated phases: Concept, Focus Area One, Focus Area Two, and Focus Area Three—meaning that Q-Learning can be simplified and optimized for smaller. more directed state selections—there is no need for the Q-Learning in *Ghost Writer* to pick new states during the learning process because the states already directly lead to one-another, meaning that the *nu* parameter, or "the length of the walk," can be removed. Rather, the state resulting from the Q-learning step will be stored to be used in work rating later before being directed to the next state. To simplify further, the AI agents, or Competitors, of *Ghost Writer* do not generate errors, have unique traits, or incorporate masteries like the player, allowing Q-learning to focus solely on the compatibility of topics, genres, audiences, and the allocation of focus points within the context of a given genre.

4.3 ReinforcementProblem

Ghost Writer implements the other necessary classes of the ReinforcementProblem and the QValueStore (Millington, 2019). The former class holds two Dictionaries: one whose keys are ints, to describe the number of the action, and whose values are Func<(float value, object data)>, which allows for the real-time rating of actions by returning a value and the necessary, calculated state data. The first Dictionary is strictly for the available actions for the Concept phase of entity work development. The second dictionary is for the Focus Areas One-through-Three, containing a Genre as its key and a List of Dictionaries—consisting of Dictionary<int, Func<(float value, object data)>—as its value. This way, the actions that can be taken for each Genre's Focus Area actions are contained within one place and can be easily referenced by using the current Genre and the stage of development.

4.4 QValueStore and QLearner

The QValueStore class is nearly the same as Millington's implementation, but with some slight differences in using structs and HashSets both for optimization purposes and programmatic clarity. Then, there is the QLearner class, whose only purpose is to run the algorithm and return the best action by setting and retrieving values within the QValueStore.

4.5 LearnedBrain and RandomBrain

The LearnedBrain class inherits from an abstract parent class denoted as CompetitorBrain. Another child is the

RandomBrain, which will be used to contest the LearnedBrain in data trials. The LearnedBrain takes care of the initialization process for each Competitor. Each Competitor, for the sake of testing and performance, is given five topics, three genres, and all of the four audience types. As they are initialized, they store the functions to test the compatibility of each Genre-Topic match, each Topic-Audience match, and each Genre-Focus Area match within the ReinforcementProblem's Dictionaries, and initialize their unique tailored parameters for Q-Learning. Then, when they are called to make a work, they go in order of development: Concept, Focus Area One, Focus Area Two, and Focus Area Three, storing each found best action to reference later for the final rating of their work.

5 Evaluation

The learned AI will be evaluated by being compared to random agents, which are so commonly used in other management-tycoon games. Random agents are those who select randomly from their given options. These differences come from the LearnedBrain and the RandomBrain. While the LearnedBrain will use Q-Learning, finding the best Concept and Focus Area combinations from their available options, the RandomBrain will select randomly from their available options. There are three sets of data: Dataset 1 and 2 uses six learned agents and one random agent. The six learned agents each have a strong, weak, and middle-ground trait, which are reflected in their Q-Learning parameters *alpha*, *gamma*, and *rho*, which are clarified to their more legible labels: Learning Factor, Discount Factor, and Exploration Factor. Dataset 1 holds ten trials, pitting the agents against each other, each using three iterations of Q-Learning per step before retrieving a final action. Dataset 2 uses the same agents, but increases the number of iterations to 50 per Q-Learning step over five trials. Dataset 3 then attempts to remove all inconsistencies (such as better or worse Genre-Topic pools) by having all the agents pick from the same available genres and topics, but they each have different tailored parameters. This allows for a study in the tailored parameters, denoting how each parameter affects an agent and which combinations will produce stronger or weaker learned agents. Each dataset ranks an agent's performance by the average score of their works and their total sales by the end of three in-game years.

6 Results

6.1 Dataset 1 and 2

6.1.1 Agents

Seven agents were described in Dataset 1 and 2:

- Edgar Allan Poe: Learning Factor 1; Discount Factor 0.2; Exploration Factor 0.7
- Hunter S. Thompson: Learning Factor 1; Discount Factor 0.7; Exploration Factor 0.2
- M. L. Wang: Learning Factor 0.2; Discount Factor 1; Exploration Factor 0.7
- Thomas Pynchon: Learning Factor 0.7; Discount Factor 1; Exploration Factor 0.2
- R. F. Kuang: Learning Factor 0.2; Discount Factor 0.7; Exploration Factor 1
- V. E. Schwab: Learning Factor 0.7; Discount Factor 0.2; Exploration Factor 1
- Nathaniel Hawthorne: Unlearned; random selections

Each agent wrote a Short Story as their type of work, meaning they all took the same amount of time to produce a work, resulting in 12 works for all agents by the end of the third in-game year.

6.1.2 Dataset 1 (3 Iterations)

Ten trials were held within Dataset 1, each lasting from January 01, 2024 to January 01, 2027 using the in-game Calendar. For each agent, the Q-Learning algorithm iterated three times per step. Furthermore, each "Average sales" value is rounded to the nearest whole number.

- Edgar Allan Poe: Average score of 87.98; Average sales of 108,360
- Hunter S. Thompson: Average score of 73.88; Average sales of 61,651
- M. L. Wang: Average score of 74.67; Average sales of 62,082
- Thomas Pynchon: Average score of 65.60; Average sales of 44,319
- R. F. Kuang: Average score of 78.47; Average sales of 78,610
- V. E. Schwab: Average score of 85.33; Average sales of 98,769
- Nathaniel Hawthorne: Average score of 56.28; Average sales of 27,242

Each learned agent performed better than the random agent by a fair margin. The lowest performing learned agent (Thomas Pynchon) still performed ~163% better than the random agent (Nathaniel Hawthorne).

6.1.3 Dataset 2 (50 Iterations)

Five trials were held within Dataset 2, each lasting from January 01, 2024 to January 01, 2027 using the in-game Calendar. For each agent, the Q-Learning algorithm iterated 50 times per step. Furthermore, each "Average sales" value is rounded to the nearest whole number.

• Edgar Allan Poe: Average score of 98.97; Average sales of 160,681
• Hunter S. Thompson: Average score of 95.52; Average sales of 140,912
• M. L. Wang: Average score of 72.70; Average sales of 56,290
• Thomas Pynchon: Average score of 74.20; Average sales of 67,561
• R. F. Kuang: Average score of 85.87; Average sales of 99,219
• V. E. Schwab: Average score of 96.65; Average sales of 146,221
• Nathaniel Hawthorne: Average score of 26,533; Average sales of 26,533

Building on the trend seen in Dataset 1 (6.1.2), the learned agents significantly outperformed the random agent. However, with more iterations, each learned agent, with the exception of M. L. Wang performed higher than their three-iteration version. The tailored parameters for M. L. Wang and Thomas Pynchon (high discount, weaker learning, and exploration) explain why their performance did not improve significantly compared to the three-iteration version. Their iterations were less effective than those of peers with stronger learning or exploration parameters, as they were more likely to converge on a particular Q-value.

6.2 Dataset 3

6.2.1 Agents

Unlike the agents seen in Datasets 1 and 2, which all had different topics and genres chosen for them, the agents within Dataset 3 were all based on the Edgar Allan Poe agent from Datasets 1 and 2. This allowed each agent to be uniform in their topics and genres, removing the possible inconsistencies of unbalanced choices from the equation and solely focusing on the effects of the tailored parameters.

• Edgar Allan Poe (All Low): Learning Factor 0.2; Discount Factor 0.2; Exploration Factor 0.2
• Edgar Allan Poe (D): Learning Factor 0.2; Discount Factor 1; Exploration Factor 0.2
• Edgar Allan Poe (DE): Learning Factor 0.2; Discount Factor 1; Exploration Factor 1
• Edgar Allan Poe (E): Learning Factor 0.2; Discount Factor 0.2; Exploration Factor 1
• Edgar Allan Poe (L): Learning Factor 1; Discount Factor 0.2; Exploration Factor 0.2
• Edgar Allan Poe (LD): Learning Factor 1; Discount Factor 1; Exploration Factor 0.2
• Edgar Allan Poe (LDE): Learning Factor 1; Discount Factor 1; Exploration Factor 1
• Edgar Allan Poe (LE): Learning Factor 1; Discount Factor 0.2; Exploration Factor 1

Each agent wrote a Short Story as their type of work, meaning they all took the same amount of time to produce a work, resulting in 12 works for all agents by the end of the third in-game year.

6.2.2 Dataset 3 (10 Iterations)

Five trials were held within Dataset 1, each lasting from January 01, 2024 to January 01, 2027 using the in-game Calendar. For each agent, the Q-Learning algorithm iterated 10 times per step. Furthermore, each "Average sales" value is rounded to the nearest whole number.

• Edgar Allan Poe (All Low): Average score of 67.88, average sales of 45,831
• Edgar Allan Poe (D): Average score of 63.27, average sales of 40, 456
• Edgar Allan Poe (DE): Average score of 76.50, average sales of 69,958
• Edgar Allan Poe (E): Average score of 77.13, average sales of 70,865
• Edgar Allan Poe (L): Average score of 87.49, average sales of 104,160
• Edgar Allan Poe (LD): Average score of 59.28, average sales of 32,812
• Edgar Allan Poe (LDE): Average score of 61.87, average sales of 35,915
• Edgar Allan Poe (LE): Average score of 95.70, average sales of 141,939

By the end of the five trials, it was apparent that those that had a high discount factor (D, DE, LD, and LDE) performed worse than those with a high learning or exploration factor (as long as those agents didn't also have a high discount factor; E, L, LE). As the LE agent further emphasizes, agents with both a high learning and exploration factor, with a lower discount factor, will demonstrate increasingly high values in terms of score and sales, as also shown by Edgar Allan Poe and V. E. Schwab in Datasets 1 and 2. The discount factor disparity in Dataset 3 further provides clarity to why M. L. Wang and Thomas Pynchon—both agents with a high discount factor—were outperformed by most other agents with lower discount factors in Dataset 2: despite the increased number of iterations, the other agents with lower discount factors were allowed to expand away from the max Q-value state.

7 Conclusions

After running multiple trials within the three datasets, it can be concluded that any learned agent will outperform any random agent greatly, despite parameter tuning. Though random agents can be used to help balance the cast of competitors, this can also be done by further tuning the tailored parameters of each individual agent, providing a base "intelligence" floor for each Competitor, raising the overall difficulty and competition of the game.

7.1 Parameter Tuning

For faster learning, the learning factor and exploration factor should be set to a higher number. The rate of which

controls the speed of the learning also depends on the amount of iterations per step. For slower learning, like in Dataset 1, less iterations will work better. Lower numbers will work better for games that have a long gameplay lifespan, such as management-tycoon games, which can span years-to-decades of in-game time. For *Ghost Writer*, it seems that somewhere between the range of 3-5 iterations per step suits the overall lifespan. For agents that would want to stick to certain scores—which can be more realistic for authors as they repeat certain themes, topics, and genres—a higher discount factor can be used with lower exploration; however agents with a high discount value seem to always underperform compared to other agents with lower discount values, even if their other parameters are equal or greater.. The ultimate agent will have high learning and exploration factors and a low discount factor with many iterations, as seen with Edgar Allan Poe and V. E. Schwab in Dataset 2, however, these super-powered agents might be better suited to harder difficulties, where having such a large, initial boost in sales makes tactical sense within the realm of the game. All-in-all: higher learning results in stronger agents that will prove to have higher sales sooner; higher discount results in scores more alike to a given score (decided by the max Q-value); higher exploration results in a quicker expansion of AI knowledge; and higher iterations will only empower the ability of the learning and exploration factors.

7.2 Why?

After all of this, it may seem easier to have agents that are tailored to one specific range of values, however, these types of patterns are more static and can become increasingly repetitive while also proving to be unrealistic. Authors in real life will not always score a best-seller, and not many of them will do that within their first or second book; they will learn as they continue to work. *Ghost Writer*'s agents will provide a more realistic and more competitive environment for the player as AI agents learn how to play alongside the player, but can be tailored to understand the player at different intensity levels. Furthermore, these AI agents are extremely customizable and the variance in their tailored parameters can result in different behaviors. For instance, higher discount values can cause the learned agents to stick towards a specific Topic and Genre, which is realistic for certain authors who have a preferred style; higher exploration values mimics authors who have a desire to expand their stories to different realms and contexts; higher learning mimics authors who might have a knack for understanding the art of narrative and accelerate in the field.

8 Future Work

The learned agents displayed here within *Ghost Writer* are still limited. They have yet to incorporate mastered topics and genres, which the players might have already figured out, nor do they consider trending genres or themes. The *Ghost Writer* learned agents are bare-bones in terms of internal data, and unpolished in terms of parameter tuning. Future work can explore economic reactions within agents, such as needing to sacrifice possibly a bigger scale project for a smaller work that will help them pay for the current month (resulting in the incorporation of work types in decision-making), which can be explored through Goal-Oriented Action Planning (GOAP). Furthermore, incorporating all aspects of the written work (such as errors and masteries) can improve the Q-learned agents, as that data within the state can drive to more complexities (for example, higher discount—which performed the worst out of all the tailored parameters—could immediately become viable with the use of masteries). Overall, there are many avenues for the learned agent behavior in *Ghost Writer* to improve beyond learning the compatibility between topics, genres, audiences, and focus points and future work will explore reactionary and planned behavior to the Competitors of *Ghost Writer*.

9 References

[1] Millington, I. 2019. *AI for Games*. Boca Raton, FL: Taylor & Francis Group.