Generation of Diverse Stages in Turn-Based Role-Playing Game using Reinforcement Learning

SangGyu Nam and Kokolo Ikeda

2019 IEEE Conference on Games (CoG)

s1290102 Hajime Fukai

Introduction

Overview of this paper

- This paper investigates the application of **Reinforcement Learning** for **Procedural Content Generation**
  - Instead of other methods such as Variational Auto-Encoder (VAE) or Generative Adversarial Networks (GAN)
  - Reinforcement Learning does not require the training data
    - Applicable even when the training data is not ready; it is often the case when developing a new game
- They selected **turn-based RPG** as their primary target
- They aimed to generate **diverse stages**

Definition of Turn-based RPG

What is Turn-based RPG?

- **Examples of turn-based RPGs**: Dragon Quest, Final Fantasy, Pokémon
  - Each game features its own unique system, but most of them have the same mechanism in common
- A majority of RPG has stages consist of **battle sections**, and **non-battle sections**
  - In a battle section, the player and the enemy teams fight against each other
  - In a non-battle section, the player can cure damaged characters, or buy items and equipment

Definition of Turn-based RPG

What is Turn-based RPG?

- The condition of player characters, such as items and health, is inherited after each battle
  - Players have to think about resource management
  - For example, players have to avoid wasting their items against weak enemies
- Players have to think about their strategy and make decisions for each section
  - For example, a player may decide to advance more, or withdraw to town
  - This study aims to generate stages that require such decisions

Related work

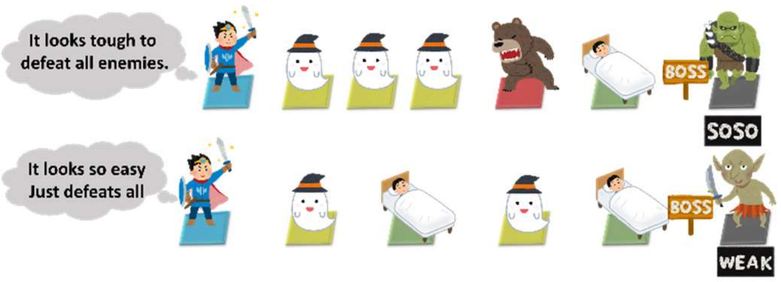
Other methods for PCG

- Search-based PCG
  - Generate-and-test algorithm
    - Combination of “selection by an evaluation function” and “generating content using any PCG methods”
  - Typically involves a Genetic Algorithm. GA is slightly slow and possibly generates similar content
- Procedural Content Generation via Machine Learning (PCGML)
  - Requires training data
  - Difficulty in data categorization

Research Platform

A simple strategy game

- This study focuses on stages with battle section and non-battle recovery section
- They implemented their own platform
  - A stage consists of n sections; each section is one of a **battle**, **recovery**, or **boss battle**
  - Both player team and enemy team consist of a single character
  - Three types of characters: a player, an enemy, and a boss
  - The speed parameter of the players is always greater than that of the enemy

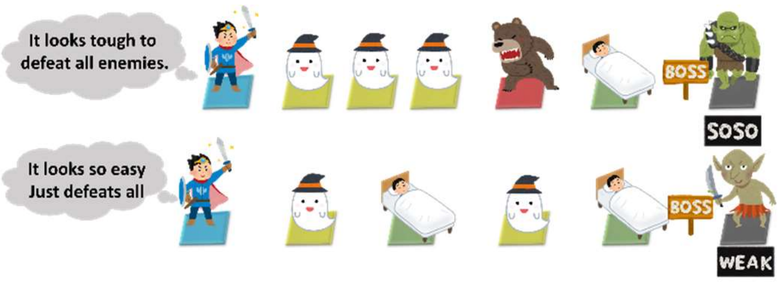


Examples of various stage structures

Research Platform

A simple strategy game

- They implemented their own platform
  - The player can take two actions: **attack** or **retreat**
    - Attack damage depends on ATK parameter of the player
  - When the player retreats from the battle, the player character loses 15% of HP, and proceeds to the next section
  - When the player wins the battle, the player character gains 10% of ATK
- A stage has 2^{m-1} possible strategies, where m is the number of battles

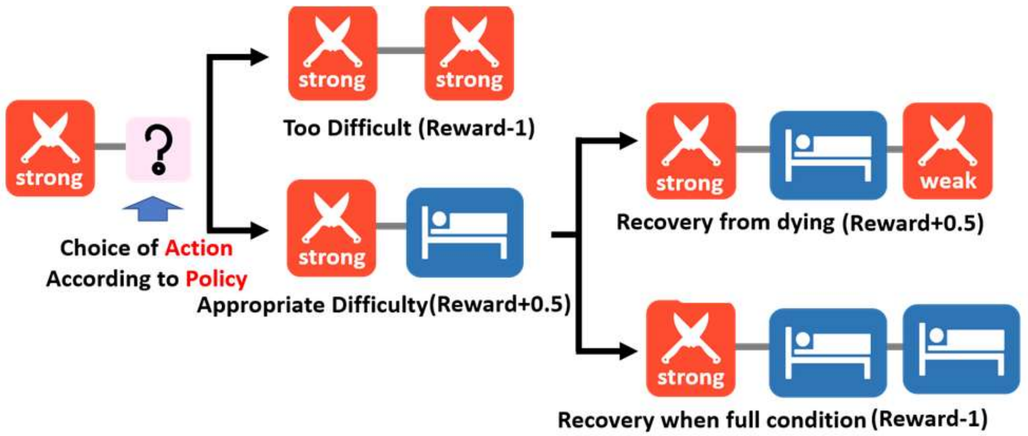


Examples of various stage structures

Approach to stage generation

Markov Decision Process

- MDP Formulation
  - States: All stages including incomplete stages
  - Action: Parameter manipulation of the incomplete stage
  - Goal state: The completed stage
  - Reward: Evaluation of the stage
- Incomplete random initial stage
- Stochastic noise policy



MDP process of generating stages

Approach to stage generation

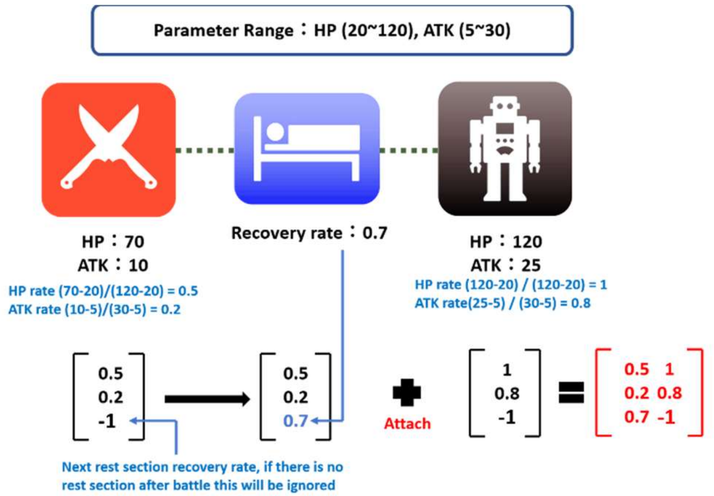
Reinforcement Learning

- RL methods
  - **Deep Q-Network (DQN)** was first selected
    - They observed some limitations in PCG
    - Suitable for a discrete, lower-dimensional action space
      - It is difficult to generate several parameters at once
  - **Deep Deterministic Policy Gradient (DDPG)** was selected next
    - DDPG can deal with a high-dimensional continuous action space

Approach to stage generation

How stages are represented by matrices

- Stage matrix representation
  - Let n the number of sections of the stage and the stage consists m battle sections
  - Let each battle section has P_b parameters and each recovery section has P_r parameters
  - Then the size of a stage matrix is $(P_b + P_r) \times m$
- Each column represents $\begin{bmatrix} \text{HP rate} \\ \text{ATK rate} \\ \text{Recovery rate} \end{bmatrix}$



Example of converting the battle-recovery-boss stage to the stage matrix

Approach to stage generation

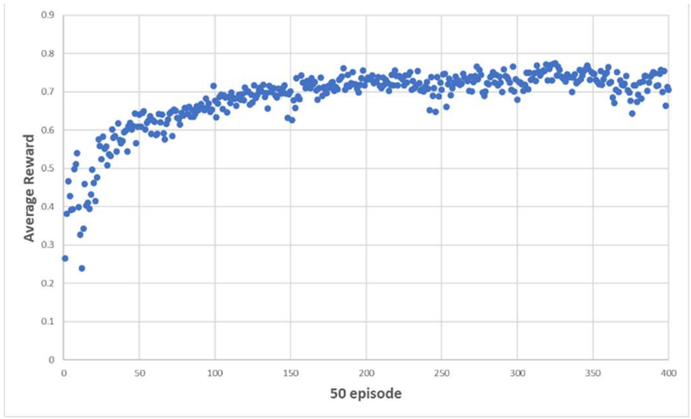
How stages are evaluated

- Stage evaluation
  - Difficulty-based evaluation
  - The difficulty is confirmed by searching all the 2^{m-1} strategies
  - “Winning rate”: The percentage of strategies that can defeat the boss, among all the strategies
  - The target winning rate was assigned to be 30% in this study
  - Higher evaluation if the winning rate is close to the target rate

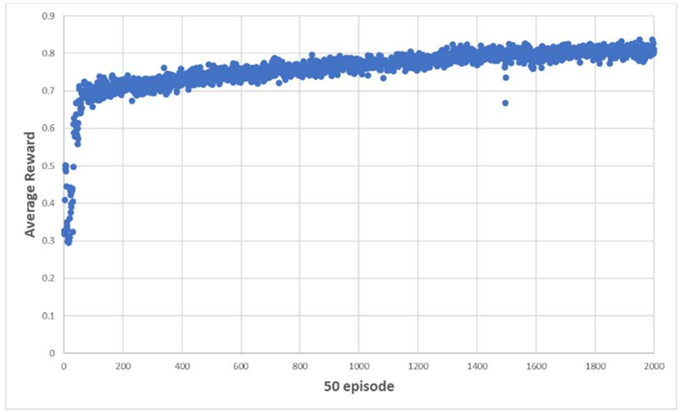
Experiment: Stage Generation

Performance of DQN and DDPG

- This experiment is performed to generate stages using DQN and DDPG
- Both methods could generate stages with high evaluation value
- DDPG gained more reward than DQN



Average reward for 50 episodes (total 20000 episodes) of stage generation using DQN

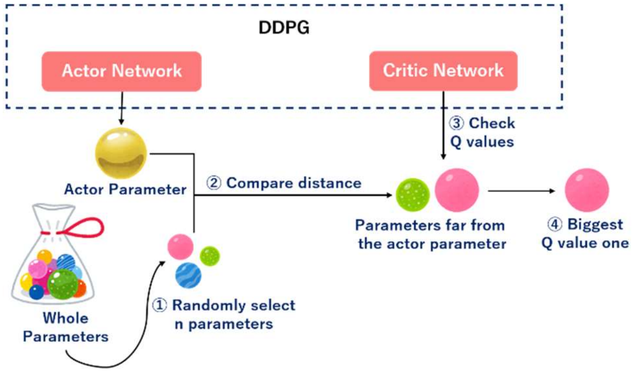


Average reward for 50 episodes (total 100000 episodes) of stage generation using DDPG

Experiment: Diverse Stage Generation

Stochastic noise policy

- This experiment was performed only with DDPG
- Stochastic noise policy is employed
  - The target columns are decided
  - n random parameters are selected, for each target column
    - If the distance between a parameter and DDPG's actor parameter is less than d, the parameter is discarded
    - Generate a column with the parameter with the highest Q value
  - m stages are generated in total



Flow of selecting the noised action in the stochastic noise policy

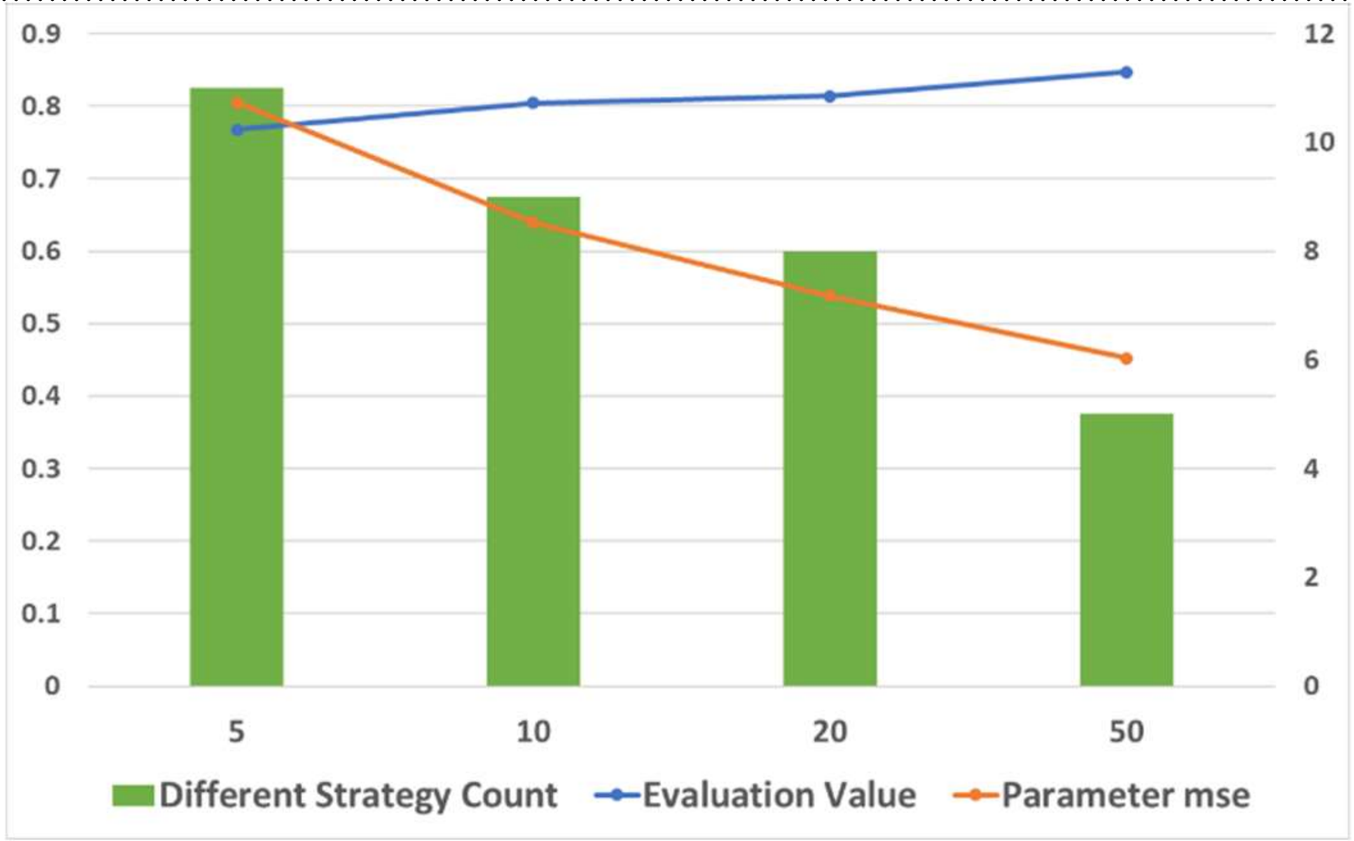
Experiment: Diverse Stage Generation

Examples of the generated stages

Stages generated by the actor policy and stochastic noise policy (sections 3, 4, 7, and 8 are noised sections). Actor one is evaluated as 0.928661, the other is evaluated as 0.894968. The stage parameter mse between the two stages is 0.3525, and the different numbers of valid strategies is 5

Experiment: Diverse Stage Generation

Experiment result



Average reward and the average number of different valid strategies as well as the average parameter mse of stages generated by the stochastic noise policy when n = 5, 10, 20, and 50; m is 50; and d is 0.2. The target noise column are 1st and 6th

Thank you for your attention