

CMSC 691 Interactive Fiction Project Update 3

Shawn Bray, Josh Li, Reece Robertson

November 2024

1 Project Title

The title for our project is the *Dice Roll Action Generating Operational Network (DRAGON)*.

2 System Diagram

The system diagram for our project is as follows:

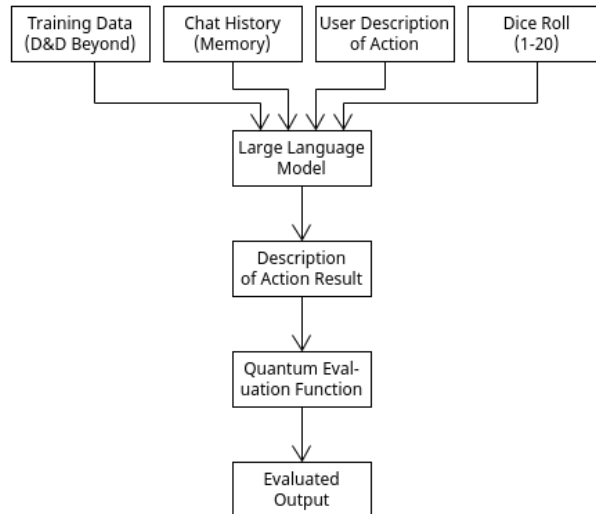


Figure 1: *DRAGON* system diagram.

3 Team Members

- **Shawn Bray.** Shawn led out in the project ideation and planning stages. He synthesized the ideas of different group members into an executable project, and has research papers related to our project. He will help finish and validate the code, and will contribute to the final presentation.
- **Josh Li.** Josh took point on the project proposal document. He also wrote all of the code demonstrated in the video linked below. He has worked hard to implement and debug the game framework. He recently added additional features to the game framework, such as a memory system, and will contribute to the final presentation.
- **Reece Robertson.** Reece contributed to the research for the project proposal, and he wrote the text for both project updates. He also filmed the video linked below. Reece is also experimenting with a quantum function to evaluate the quality of the output of the model, which may be incorporated into the final product demonstration. He will help finish and validate the code, and will contribute to the final presentation.

4 LLM Use Statement

The large language model (LLM) in this code is Llama 2. Specifically, it is:

Llama 2-7b-chat: This is the 7-billion parameter version of Meta’s Llama 2 model, fine-tuned for chat-based tasks.

5 Methodolgy

This algorithm is designed to generate a narrative response for a interactive game based on a player’s action and the game’s history. It starts by rolling a dice to introduce an element of randomness to the outcome, where a roll of 10 or higher results in a "success" and anything lower leads to a "failure." This outcome is then used to guide the narrative generated by a language model.

Algorithm 1 Generate Game Response with Interactive Dice Roll

Require: Game history $history$, Player's action $action$

Ensure: Generated narrative for the game response

```
1: function ROLLDICE
2:   return random integer  $d$  where  $d \in [1, 20]$ 
3: end function
4: function GENERATEGAMERESPONSE( $history, action$ )
5:   Combine the last three entries of  $history$  to form  $history\_context$ 
6:    $history\_context \leftarrow \text{JOIN}(history[-3 :])$ 
7:   Roll a dice
8:    $dice\_result \leftarrow \text{ROLLDICE}$ 
9:   if  $dice\_result \geq 10$  then
10:     $outcome \leftarrow \text{"success"}$ 
11:  else
12:     $outcome \leftarrow \text{"failure"}$ 
13:  end if
14:
15:   Construct a prompt  $prompt$  for the LLM
16:    $prompt \leftarrow \text{"You are narrating an interactive fiction game. Given game's history and player's action, describe what happens next in 2-3 sentences. Player's action resulted in a \{outcome\}. Focus only on immediate new situation. Game history: \{history\_context\} Player action: \{action\} What happens next:"}$ 
17:
18:   Tokenize the  $prompt$ 
19:    $model\_inputs \leftarrow \text{TOKENIZER}(prompt, \text{return\_tensors})$ 
20:
21:   Generate the model output
22:    $output \leftarrow model.GENERATE(model\_inputs, \text{max\_length}, \text{temperature})$ 
23:
24:   Decode the generated output
25:    $new\_situation \leftarrow \text{TOKENIZER.DECODE}(output, \text{skip\_special\_tokens}) \text{len}(prompt)$ 
26:
27:   return  $new\_situation$ 
28:
29: end function
```

The main function of the algorithm first combines the last three entries of the game history to create a context for the language model. It then constructs a prompt that describes the game history, the player’s action, and the result of the dice roll. This prompt is tokenized and used as input for a pretrained language model that generates the next part of the game narrative, which is then decoded and returned as the game’s response.

Our approach leverages the capabilities of language models to create dynamic and contextually relevant game responses, making each interaction unique. The use of dice rolling introduces variability, enhancing the gameplay experience by adding an element of chance. Overall, this algorithm efficiently combines random elements with advanced language generation techniques to create engaging and interactive game narratives.

6 Results

We find as a proof of concept that the LLM model design can successfully generate a option at each stage of the narrative for the user to progress logically from start to finish. From the examples below in each figure, from the validation of each action and response it is shown to correctly translated through each process.

```
Welcome to the AI Generated Adventure!
```

```
Your situation: You are creating an interactive fiction game. Describe the initial setting and situation the player finds the mselves in. Keep it to 2-3 sentences.
```

```
The player finds themselves in a dark forest, surrounded by towering trees that block out most of the sunlight. The air is th ick with the scent of damp earth and decaying leaves, and the only sound is the distant howling of wolves. As they look aroun d, they notice a small clearing with a glowing path of light leading through the trees.
```

```
What would you like to do? (type 'quit' to end)  
> I fight the wolves with my sword.
```

Figure 2: The player charges towards wolves with their sword, but they are overwhelmed as the wolves retaliate aggressively.

```
As you charge towards the wolves with your sword raised, they turn and run at you, their eyes glowing in the dark. You swing your sword and manage to hit one of them, but there are too many of them and they overwhelm you, their jaws closing in on yo u.
```

```
What would you like to do? (type 'quit' to end)  
> I use the sword to defend myself.
```

Figure 3: The wolves continue attacking the player, who struggles to defend against their persistent assaults.

```
The wolves continue to attack you, their bites and claws raining down on you. You struggle to keep them at bay, but their she er number is starting to wear you down.
```

```
What would you like to do? (type 'quit' to end)  
> I shout at them, swing my sword meanwhile, hoping to scare them.
```

Figure 4: The wolves momentarily pause their attack as their leader steps forward and challenges the player with a growl.

```
The wolves stop their attack, momentarily intimidated by your boldness. However, their leader, a massive gray wolf with piercing eyes, steps forward and growls menacingly, "You dare to challenge us? You are no match for our pack!"

What would you like to do? (type 'quit' to end)
> quit

Thanks for playing!
```

Figure 5: The game introduces the player to a dark forest setting, emphasizing the eerie atmosphere and the distant howling of wolves.

Our project combined probabilistic decision-making with natural language generation to create a dynamic, interactive fiction framework. We used a dice roll mechanic to introduce randomness, determining the success or failure of players' actions. Our approach made the narrative more engaging. Using advanced language models, the system generated context-aware game responses based on a rolling memory of the game history and player inputs. The game started with an AI-generated setting that immersed players in a fictional world and evolved as they interacted with the environment through text-based commands.

The results show the potential for mixing rule-based mechanics with machine learning to create open-ended, adaptive storytelling experiences. Our framework successfully demonstrated how these technologies could work together to craft immersive narratives. For future work, we could focus on optimizing memory size management and refining the outcomes of critical success or failure to add more depth to the story, making it even more complex and engaging. Overall, our project shows that it can lead to a more captivating interactive fiction experience.

7 Conclusion

Our approach ensures that the narrative stays engaged through the users experience with the game. The proposed framework opens up to potentially new opportunities for future improvements, like better memory management and refining the outcomes of critical successes or failures to make the story even more complex. Overall, the project was fun and interesting.

8 References:

- Link to codebase: https://github.com/josh11-umbc/cmsc691_term_project.
- Link to video: <https://umbc.webex.com/umbc/ldr.php?RCID=bfa832ea69d0d19bad7b1fa46d1b0476>.