

A Natural Language Text to Role-Playing Game Animation Generator

Caio de F. Oliveira
Federal
University of Ceará
cfoviana@alu.ufc.br

Artur O. R. Franco
Federal
University of Ceará
arturfranco@ufc.br

Wellington Franco
Federal
University of Ceará
wellington@crateus.ufc.br

José G. R. Maia
Federal
University of Ceará
gilvanmaia@virtual.ufc.br

Abstract

The visual elements are important for computer games. In particular, cinematic animations and cutscenes play a key role in supporting the narrative on the popular *Computer Role-Playing Game* (CRPG). Despite being based on the narrative, however, such animations usually consume significant development time, budget, and effort from animators. To enhance this aspect of the game development process, in this work, we propose a novel semiautomatic approach for generating animations for *Role-Playing Game* (RPG) based on *Natural Language Processing* (NLP) over narrative texts. We conducted a case study to validate our strategy in which, using three well-known themes, text was generated by an artificial intelligence system, and animations were subsequently created based on these texts.

1 Introduction

CRPG are narrative-heavy and often feature visual elements that are intended to convey the narrative (Barton and Stacks, 2019). In fact, various animations and character dialogues in the story usually are the focus of work since the first CRPG of *Dungeons & Dragons* in 1982 (Barton and Stacks, 2019). The aesthetic perception of what constitutes a game animation and what does not is defined by its framing and reframing. This structure places emphasis on the movement and the scene of characters (actors), often taking into account what would be outside the cinematic frame (Santaella, 2009).

The type and production cost of animations vary depending on the game engine, techniques, aesthetics, and media used. Still, most game development teams demand animation professionals responsible for producing quality cutscenes with the proper use of a sense of cinematography (Cooper, 2021). However, thinking and implementing these animations can consume significant development time, budget, and resources.

In this paper, we propose an approach to processing Portuguese sentences and generating corresponding animations for CRPG, whose focus is the transformation of concise English sentences into animations (Oshita, 2010). Titles can benefit from our approach to fill the animation gap or even avoid presenting poor content given animators can better focus their efforts on fine-tuning generated animations.

The main challenge in this development is to handle a wide range of verbs generically since there are numerous verbs, and creating a specific implementation for each one is infeasible.

We implemented and evaluated our approach using the *RPG Maker MV* (RMMV) engine (Cooper, 2021) (Sheldon, 2022) as the main target platform. Regarding the outcomes of this implementation, it involved the generation of animations through the processing of brief sentences authored by an Artificial Intelligence system, which were subsequently customized by the user to ensure the production of reliable and accurate results.

This paper delves into several components related to actions (Kearns, 2017; Hayton et al., 2020), including their registration in the database and subsequent analysis, all of which will be extracted from sentences and transformed into animations.

2 Background

This work is built upon concepts of *Natural Language* (NL), especially concerning the description of actions and their conversion into graphical elements. Additionally, it is necessary to clarify some aspects of the animation process, both from an aesthetic and a technical perspective, as the latter also guides its application in the context of games.

2.1 Games and Visual

Visual appeal serves as a pivotal attraction in the realm of digital games. The focus on aesthetics pre-

dates the pixelated graphics of Atari games and is evident in the earliest generation of games, e.g., *Adventure* and *Pitfall* (Montfort and Bogost, 2020). In particular, *Adventure* played a defining role by introducing the third-person camera structure that has become a common feature in CRPG and pioneered the concept of loading and unloading sections of the game world (Montfort and Bogost, 2020).

2.2 Animation Principles and Systems

Johnston and Thomas (1981) defined the 12 fundamental principles of animation, which are: squash and stretch; anticipation; staging; straight ahead action and pose to pose; follow through and overlapping action; slow in and slow out; arcs; secondary action; timing; exaggeration; solid drawing; and appeal. Even though simplified animations are used in our work, these general principles are adopted.

The fundamental principles of animation in digital games draw inspiration from cinema and traditional animation, as comprehensively detailed by Cooper (2021). Even within the context of digital games, Disney’s 12 principles of animation remain relevant, along with the 5 principles of game animation (Hoberman, 1982).

Actual animations in games are implemented by a computer system. Within this regard, the approach advocated by Shapiro (Shapiro, 2011) simplifies the problem by segmenting animations into smaller components, referred to as *controllers*. This technique will also be employed in our work, but the division will be streamlined, as it is tailored for a 2-Dimensional RPG game with a minimal level of detail.

2.3 Action Events

According to Kearns (2017), Vendler’s four semantic classes of action events (accomplishment, achievement, state, and activity) are characterized in terms of three main distinctions: telicity, dynamism, and duration. *Telicity* refers to the property of action events having a natural finishing point. *Dynamism* refers to the property of an event or action that implies movement, change, activity, or action over time, i.e., while static events have dynamic uniform states, dynamic events involve ongoing actions or changes in state over time. Finally, *duration* refers to the period during which an event occurs or an action unfolds, i.e., a durative event occupies time.

Consequently, for the purposes of this work, we assume events must be telic, dynamic, and durative.

2.4 Natural Language Processing

Our approach uses elements of narrative as the starting point, so we shall resort to well-grounded NLP techniques. The text processing in NLP is divided into several stages, each performing a specific task, and together, they are referred to as the **processing pipeline**. An example of a typical pipeline is found in the popular SpaCy¹ library.

The model used for this work is called “pt_core_news_lg”², which is provided by SpaCy, whose pipeline consists of six components: **tok2vec**, which converts tokens into vectors; **morphologizer**, which defines the morphological classes of tokens; **parser**, which is responsible for the relationships between tokens; **lemmatizer**, whose function is to determine the basic forms of tokens; **attribute ruler**, which allows the customization of token attributes based on specific rules; and **NER**, the named entity recognizer.

3 Related Works

Hassani and Lee (2016) described the requirements for systems that convert natural language into graphical elements, which include: an engine for visualizing the generated graphical elements; a formal natural language description system; and an architecture that combines the two aforementioned requirements, which is the focus of this work.

Furthermore, these authors also categorized systems that convert natural language into graphical elements into three types: “**text-to-picture**”, which searches for images in a database that closely match the provided description; “**text-to-scene**”, which generates static images; and “**text-to-animation**”, which generates animations and is the focus of this work. Other **Text to Animation (TTA)** systems are presented in the survey by Bouali and Cavalli-Sforza (2023).

Other works, not limited to gaming but encompassing multimedia products generated from text, can be found, as exemplified by Hayton et al. (2020). In this study, a system is introduced that takes natural language input and generates a **Planning Domain Definition Language (PDDL)** model as the output of a narrative.

In his turn, Oshita (2010) introduced a framework for converting each verb in English script-like sentences into animations. However, this approach does not cover situations involving sentences with

¹<https://spacy.io/usage/processing-pipelines>

²https://spacy.io/models/pt#pt_core_news_lg

multiple subjects or phrases containing more than one verb, as illustrated in Figures 3 and 4. Our approach is similar, but we address the analysis and resolution of these specific cases. Moreover, our scope is limited to the 2D CRPG animations.

Each of these components plays a crucial role in text processing and analysis, contributing to SpaCy’s ability to perform a wide range of natural language processing tasks.

4 Development and Analysis

As stated beforehand, the sentence processing resorted to the SpaCy library for Python using a model trained for the Portuguese language. With the sentences properly processed, it was possible to generate animations for an RPG game using the RMMV engine.

4.1 Game Engine

RPG Maker is a series of game development software that enables users to create their own RPG, providing tools for character creation, map design, dialogue, events, and battle systems, making it easier to create customized RPG. The RMMV³ was chosen because (1) it exports games to multiple platforms, e.g., web browsers and Linux; and (2) game data (e.g., characters, maps, and events) are primarily stored in JSON files, making it easy to edit and configure these elements using custom, external tools. This choice not only allows for our research but also favors both experimentation and the reproducibility of results.

4.2 Custom Spacy Pipelines

As mentioned in 2.4, the Spacy tool incorporates **pipelines**. To enhance the functionality of our parser, we chose to customize certain components of these pipelines.

To start, we developed a custom pipeline to handle word gender, particularly impacting coreference, a topic we’ll delve into shortly. In a separate file, users are required to specify words exempted from their literal gender assignment. For instance, “*Chapeuzinho Vermelho*” (Little Red Riding Hood) is a female character, but due to the masculine gender of “*chapéu*” (hat), the unaltered model might erroneously assign a masculine gender to the character.

The tool lacks precision in handling cases of enclitics in verbs. To address this, we customized

the **tokenizer** pipeline to separate words and generate distinct tokens at hyphens. As a complement to this modification, we introduced a new pipeline that adjusts the **part-of-speech tag** of the newly generated token after separation. This token, representing the enclitic pronoun (“PRON”), is linked to its **head** (the verb) and is assigned a dependency tag of “obj”.

Moreover, we opted for Spacy’s **sentencizer** as the pipeline for sentence segmentation, enhancing it by introducing a new rule to allow segmentation at commas.

An isolated case that presented a distinct challenge involved verbs starting sentences. The model occasionally misinterpreted these instances, incorrectly classifying the verb as a proper noun due to its initial capital letter. Despite the proper noun label, we observed that dependents of such verbs retained certain dependency tags associated with verbs in sentences, such as subject tags. In response, we introduced a new pipeline to scrutinize these situations and adjust the labeling of these words appropriately.

In certain instances where verbs initiate sentences or subordinate clauses, the tool failed to accurately analyze dependencies, leading to the misattribution of the object as the subject of the verb. For instance, in the sentence “Chapeuzinho Vermelho caminha pela floresta quando viu o Lobo Mau” (Little Red Riding Hood walked through the forest when she saw the Big Bad Wolf), SpaCy incorrectly identifies “Lobo Mau” (Big Bad Wolf) as the subject of the verb “ver” (to see). Given that our project’s scope extends beyond the traditional subject-verb-object structure, we implemented a new pipeline to reclassify subjects (“nsubj”) occurring after the verb as objects (“obj”).

4.3 Input Data

Our approach requires the previous registration of some essential data in a database. These are related to places (locations), characters, verbs, speed adverbs, and time adverbs. Each of these elements plays a fundamental role in matching named entities as required for building and enriching the RPG experience.

Time adverbs play a crucial role in establishing the sequence of events and actions within the animation. The user needs to specify and classify each of them as *before*, *after*, or *synchronous* to define the exact chronological order of events in the animation. This empowers the user with a heightened

³<https://www.rpgmakerweb.com/products/rpg-maker-mv>

degree of control over the temporal progression of the visual narrative in the game.

Speed adverbs are pivotal in establishing the pace at which characters will move during the ongoing action. They must be classified as *fast*, *normal*, or *slow*, as this categorization offers control over the dynamics of in-game events, enabling adaptation to the context and narrative flow.

Each **place** utilized in the narrative is characterized by a unique name for identification purposes and a specific position denoted by (x, y) coordinates within the game world. This approach ensures that characters can engage with and navigate these elements in a highly immersive manner.

All **characters** referenced in the provided text must be specified, including their essential information: an initial position determined by coordinates (x, y) ; an image following the pattern established in RMMV; and an initial direction chosen from the options available in the engine.

Moreover, an additional level of flexibility exists in the capacity to establish alternative references for a given character. This feature permits the assignment of various names that direct back to the character's original definition. This dynamic favors a richer narrative and the cultivation of multi-dimensional characters within the game.

Verbs must be clearly defined in their infinitive form and classified according to the implementation possibilities offered by the animation generator. For this specific project, Table 1, the list of verbs to be considered will include *jump*, *appear*, *ask*, *avoid*, *call*, *celebrate*, *create*, *check*, *disappear*, *displacement*, *dodge*, *fight*, *find*, *follow*, *free*, *push*, *love*, *say*, *sing*, *scare*, *scream*, *see*, *search*, *take*, *touch*, *turn*, *wakeup*.

4.4 Text Processing

Our system has been designed to effectively manage sentences containing dynamic and telic verbs. So, it is imperative to include subjects, whether they are explicitly stated or implied, as the primary emphasis lies in describing actions. It is also essential that these sentences maintain conciseness and avoid the inclusion of subordinate clauses. In cases where the user intends to describe more intricate actions, it is imperative to decompose them into shorter sentences.

4.4.1 Sentence Segmentation

The Portuguese sentence segmentation performed by SpaCy primarily relies on punctuation. Never-

theless, due to the project's emphasis on actions, clause-based segmentation gains greater significance compared to the default model's approach. Consequently, in addition to SpaCy's segmentation, it must be conducted a detailed analysis of the verbs within each sentence. This in-depth analysis will yield a more precise understanding of the actions and narrative structure, enhancing the effectiveness of the generated animations.

Following sentence segmentation, it is straightforward to identify the main verb in each sentence, designated as the **root**. Nevertheless, a given sentence may encompass multiple verbs, aside from the root, as demonstrated in Figures 4 and 1, which can be accessed via the **children** relationship established among these words by the tool. A separate analysis will be carried out for each of these identified verbs.

If the element under analysis has at least one explicit subject (as in Figure 1), it will be considered as a main clause. Otherwise, if there are no explicit subjects (as in Figure 4), it will be treated as a coordinate clause, without any division, as it will share the same subject as the preceding verb.

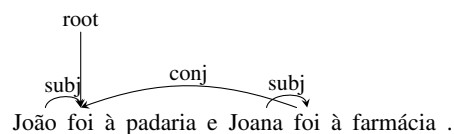


Figure 1: A sentence consisting of two clauses.

4.4.2 Structure

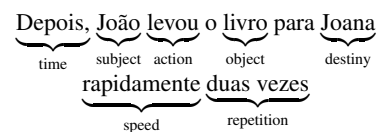


Figure 2: The structure of a sentence

The primary objective of processing each sentence is to transform it into a simplified structure that encompasses key elements, such as the **subject**, **action**, **object**, **destination**, **adverbs (time and speed)**, and the **number of repetitions**. This structuring, as illustrated in Figure 2, allows for a clearer and more organized understanding of the information contained in the sentence. It is important to emphasize that, due to the variable nature of the context, depending on the situation, certain components may be considered optional. In such cases, these will be appropriately disregarded during the

text processing process. This flexible approach enables the system to adapt to the specific needs of each sentence, optimizing the animation generation according to the elements present and relevant in each context.

4.4.3 Verbs

Verbs are at the heart of this project, serving as pivotal elements that inject action and dynamism into sentences. As shown in Section 2.3, the project’s focus is confined to straightforward sentences that encapsulate the concept of movement, specifically those featuring telic and dynamic verbs.

For every identified verb, a diligent search will be conducted within the database. In instances where a verb is not found, the SpaCy word similarity feature⁴ will be utilized. In this case, the verb in the database that exhibits the highest similarity score provided that the similarity score is greater than or equal to 70%, will serve as a substitute. However, if even this comparison method fails to find a suitable match, the verb in question will fall outside the scope of this project. Comprehensive details regarding the implemented verbs can be found in Section 4.5.3 and Table 1.

Verbal regency is a crucial element for the meaning of the generated actions. For example, in an animation, the sentence “*Joana gritou*” (Joana screamed) differs from “*Joana gritou por João*” (Joana screamed for João) as the former emphasizes emitting a sound, while the latter emphasizes calling out to the object. We handle situations where verbal regency is relevant within the animation generator itself, rather than in the parser.

4.4.4 Number of Repetitions

The feature of repeating an action will exclusively apply to **atomic actions**, which we designate as verbs that do not consist of smaller component actions within the context of this project. This specific definition will be elaborated on in greater detail in Section 4.5.3. Additionally, a basic translator has been created to receive Portuguese numbers in written form and convert them into digits.

The translator functions by utilizing a database that encompasses fully written-out units, tens, and hundreds, while also maintaining records of suffixes for millions. This database serves as the reference point to establish the corresponding translations for the numbers, which are derived from the

⁴<https://spacy.io/usage/linguistic-features#vectors-similarity>

lemmas generated through the SpaCy analysis.

When the translator identifies a unit, it will directly assign the numerical value corresponding to that unit. If the number is recognized as a ten or a hundred, the translator will increment the value by 1 and then multiply it by 10 or 100, respectively. For the specific case of the number “*mil*” (thousand), its value will be set to 1,000.

In cases where the number doesn’t fit into any of the previous categories and falls within the range of millions, as indicated by suffixes such as “*lhão*” or “*lhões*”, the suffix will undergo analysis. The final value will be determined by raising 10 to the power of the suffix and then multiplying it by 1,000.

With the specified values, the translator will reverse their order and start the addition process. Values exceeding 1,000 will be stored in a multiplication state. Otherwise, the result of multiplying the current value by the multiplication state will be added to the current total sum.

4.4.5 Subject

Subjects play a vital role in understanding sentences that describe movements or actions, as they provide the starting point for identifying and contextualizing what is happening in the sentence. During development, we identified that analyzing the subject of a sentence can have some ramifications.

The first ramification is the presence of **multiple subjects** for the same verb, as illustrated in the sentence depicted in Figure 3. In this example, “*João*” is considered the subject of the verb “*ir*”, while “*Joana*” is its conjunct. Consequently, all conjuncts of a subject must be analyzed.

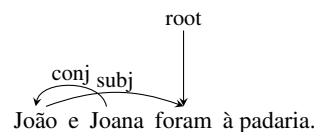


Figure 3: A sentence consisting of two distinct subjects.

The second ramification is **one subject for multiple actions**, as exemplified in Figure 4, in which “*João*” serves as the subject of the first verb “*ir*.” The second verb “*ir*” is parataxis with the first one, and “*pular*” is the conjunct of the second one. These last two verbs do not have an explicit subject, implying that we should reuse the subject from the first clause. The same applies to clauses in different sentences, as in “*João foi à padaria. Depois, pulou.*”.

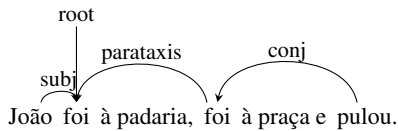


Figure 4: Coordinate clauses.

4.4.6 Correference

In the case of every personal pronoun present in the sentence being analyzed, an analysis is made to identify a reference (a noun or proper noun not within the current sentence) with the **expected gender** in preceding sentences. If a suitable reference is found, its **weight** is considered. Ultimately, the pronoun is associated with the reference or references possessing the highest weight.

The **weight** of a reference is calculated by adding 1 whenever its dependency class aligns with that of the pronoun and the gender matches the expected gender.

The **expected gender** of a reference typically matches the gender of the pronoun in most instances, except in the case of the masculine plural, which is treated differently, as it can refer to both genders as long as at least one masculine element is present.

After the analysis, should the system identify a greater number of references than expected, the sentence will be deemed ambiguous, exceeding the scope of this work. Conversely, if the number of references falls short of the expected count, the system will restart a search in the penultimate sentence. This process will iterate until the requisite number of references is achieved or until the beginning of the text is reached. In the latter scenario, if there are no references for the specified pronoun, it too lies beyond the scope of this work.

4.4.7 Adverbs

As mentioned earlier, some adverbs of **manner** determine the speed at which characters move during an action. When identified, they will be incorporated as instructions in the animation generator, including the classification of the respective adverb.

When it comes to adverbs of **time** they will be subject to distinct treatment. As previously discussed, these adverbs serve to denote the sequence of actions within a scene and encompass various classifications.

If the classification is **before**, at the end of the text analysis, the parsed sentence containing the adverb will switch places with the preceding sentence.

If the classification indicates that two or more actions occur simultaneously, i.e., **synchronous**, a marker will be inserted in the parsed sentence to enable the animation generator to process this situation correctly. At the end of the analysis, this marker will be transferred to the preceding sentence, as the animation generator needs to identify when the simultaneous action begins. Finally, if the classification is **after**, it will not influence the order of events in the text.

4.5 Animation Generation

After the analysis of the text, the corresponding parsed structure will be input into a system that converts the elements into animations in **RMMV**, utilizing the information stored in the database as specified in Section 4.3. It is important to emphasize that, despite the existence of this feature in the engine, we will not be employing multiple maps.

4.5.1 Engine

As mentioned in Section 4.1, the **RMMV** engine stores all data related to project maps and animations in JSON files. Therefore, a Python library was developed to simplify the manipulation of these files. It includes all animation commands from the engine and other essential functionalities for this project, such as map creation and event editing.

In the process of creating the animation, dedicated events for each character involved in the narrative will be added to the map. The image used for each event will correspond to that specified in the character's configuration in the database. All commands that will affect individual events will be defined in a single main event, which will remain invisible, and the user will have the ability to configure its **trigger** within the tool.

The commands utilized in this project will primarily revolve around movement route commands (**Set Movement Route**) and expression balloons (**Show Balloon Icon**).

4.5.2 Characters and Places

The essential data for characters and locations has been outlined previously, but there are still some details to be explored. It is important to note that not only animated entities will be considered characters. For example, in "*João pegou o livro.*" (João took the book), both "*João*" and "*livro*" should be considered characters since relevant information about both will be necessary.

Characters can also be linked to the target of

Classification	Verbs
jump	pular, saltar
appear	aparecer, surgir
ask	perguntar, duvidar
avoid	evitar
call	chamar, convocar
celebrate	celebrar, comemorar
create	criar, decifrar, descobrir, inventar, preparar
check	conferir, examinar, inspecionar
disappear	desaparecer, partir
displacement	adentrar, alcançar, andar, aproximar, atravessar, avançar, caminhar, chegar, correr, deslocar, entrar, fugir, invadir, ir, retornar, voar
dodge	esquivar, desviar
fight	ameaçar, brigar, desafiar, duelar, golpear, lutar
find	encontrar, achar
follow	perseguir, seguir
free	libertar
push	empurrar, espalhar, derrubar, revirar
love	amar, apaixonar, cativar, emocionar
say	conversar, contar, dizer, falar, pedir
sing	cantar, rangir, tilinta
scare	assustar
scream	berrar, gritar, rugir
see	avistar, olhar, ver, deparar, reparar
search	buscar, perder, procurar
take	levar, pegar, puxar
touch	agarrar, desdobrar, manipular
turn	girar, rodar
wakeup	acordar, despertar

Table 1: Verb classifications and their implementation.

an action, as in the sentence “*João deu o livro à Joana.*” (João gave the book to Joana). However, the target can also be a **location**, as in “*João levou o livro à padaria.*” (João took the book to the bakery). In the latter case, only location information will be needed for the element “*padaria*” (bakery), and therefore, it will be considered a **location**.

4.5.3 Action

For each verb classification mentioned in Section 4.3, an animation will be implemented. It is important to note that one action can consist of a single engine command, an **atomic action**, such as the verb **pular** (to jump), which executes the **Jump** command, or it can involve multiple commands, as in the case of the action **ir** (to go), which is carried out through several steps, each of which is a command, to reach the specified destination.

This approach ensures that any verb involving movement is handled consistently, not limited to those classified as “displacement”, which significantly reduces the implementation costs by allowing for the reuse of work across multiple actions.

4.5.4 Movement

An essential aspect to consider is movement algorithms. Pathfinding algorithms fall outside the scope of this project. As a consequence, for all types of movement in the generated animation, collision handling will not be taken into account. As a result, when designing the map, users should be mindful that the inclusion of collision elements may potentially lead to complications during the generation process.

We also ensure that the event being moved gets as close as possible to its destination without overlapping it. Each part of the movement (step) is achieved using the **Move Right** and **Move Left** commands for the horizontal axis (x) and **Move Down** and **Move Up** for the vertical axis (y).

4.5.5 Subject and Object

Subjects and objects are interactive elements within the map. As previously explained, for every interactive element, be it animated or inanimate, present in the narrative, a corresponding **event** with stored characteristics in the database will be created. All commands that pertain to these character events will be incorporated into the main event.

4.5.6 Destination

As previously stated, a destination can either be a character or a location. However, for the animation generator, this distinction is inconsequential as it only necessitates the location information. Therefore, a destination will include the final position attribute in the action, enabling the execution of movement algorithms.

4.5.7 Adverb of Speed

Within RPG Maker events, there are numerous configuration options, one of which is event movement speed, offering a choice of six options. For this project, we will exclusively focus on the three middle-speed settings, namely **x2 Slower**, **Normal**, and **x2 Faster**, aligning with the classifications described in 4.3.

4.5.8 Time Adverbs

Another configurable option in **RMMV** events is the **wait** feature, which ensures that other events cannot occur simultaneously with the one that started first. In essence, to initiate another event, the previous one must first complete its execution.

For **after** and **before** classifications, no further changes are necessary, as the required modifica-

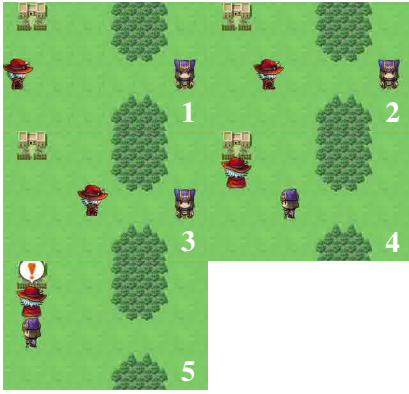


Figure 5: Animation generated by the sentence from “Little Red Hiding Hood”.

tions have already been applied by the parser. Only the wait option will be enabled.

In **synchronous** cases, concurrent execution is crucial for all events involved in the action, except for the last event, which must prevent other actions from blending into the narrative. To achieve this, the wait configuration should be enabled for all events derived from a series of simultaneous sentences, except for the last one.

5 Results

To validate the system, we utilized some tools for generating text, such as ChatGPT⁵, Bard⁶ and LuzIA⁷, within the context of the children’s story “Little Red Riding Hood” and the literary works “The Little Prince” by Antoine de Saint-Exupéry, “Peter Pan” by J. M. Barrie and “Journey to the Center of the Earth” by Jules Verne. A result for each context can be seen in Table 2, while the complete dataset of results can be accessed through the itch.io platform⁸ or GitHub page⁹.

RMMV allows the creation of plugins in JavaScript to add custom features to the created game. The developed game, when exported for browser execution, operates through the HTML canvas component. We developed a plugin that captures the game canvas at arbitrary frame intervals, storing these generated images on disk. After running the plugin, we curate the dataset by selecting the most distinctive images from the collection.

⁵<https://chat.openai.com>

⁶<https://bard.google.com>

⁷<https://luzia.com/>

⁸<https://caiofov.itch.io/animation-generator-dataset>

⁹<https://caiofov.github.io/AnimationGenerator-Dataset-PROPOR2024/>

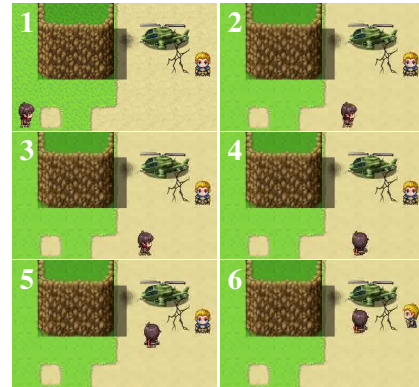


Figure 6: Animation generated by the sentence from “The Little Prince”.

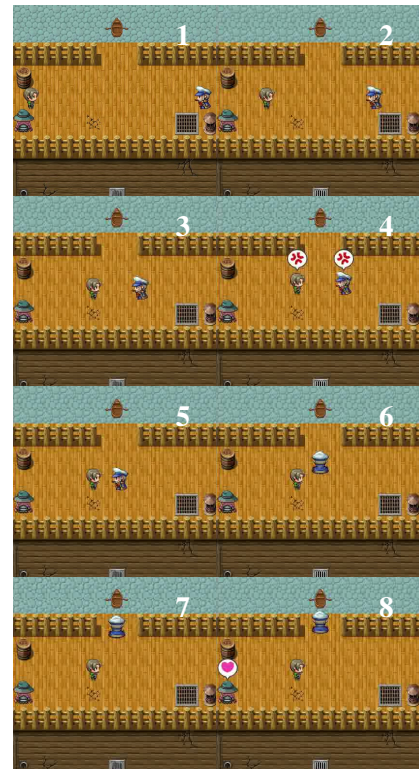


Figure 7: Animation generated by the sentence from “Peter Pan”.

Context	Little Red Riding Hood	The Little Prince	Peter Pan
Prompt	Write a paragraph with simple sentences containing dynamic and telic verbs without subordination, describing the encounter between Little Red Riding Hood and the Big Bad Wolf.	Write a paragraph with simple sentences containing dynamic and telic verbs, without subordination, to describe the encounter between the Little Prince and the pilot, based on the story "The Little Prince".	Compose a paragraph comprising concise sentences that portray the encounter between Captain Hook and Peter Pan. Use dynamic and telic verbs while avoiding subordination.
Generated text	“Chapeuzinho vermelho adentrou a floresta. Ela avistou o lobo mau. Ela fugiu, ele a perseguiu. Ela gritou, ele a alcançou.”	“O Pequeno Príncipe chegou ao deserto. Viu o avião do piloto. Correu em direção a ele. O piloto o viu. Sorriu para o Pequeno Príncipe. O encontro foi especial.”	“Capitão Gancho confrontou Peter Pan. Eles duelaram ferozmente. Espadas brilharam no escuro. Saltos, esquivas, golpes. Peter venceu, fazendo Gancho fugir. Neverland vibrou com a vitória de Peter.”
Result	Figure 5	Figure 6	Figure 7

Table 2: Input and outputs from ChatGPT.

6 Conclusions

This work introduces a model for generating animations from natural language texts. We successfully presented a functional model with telic sentences for a range of terms that can be utilized to describe actions, leveraging NLP tools and integrating them with a widespread commercial RPG engine.

The utilization of databases with descriptions provided by volunteers can be valuable in identifying areas for improvement and reinforcing the model’s validity. As for future work, the potential of integrating deep learning technologies at various stages of content processing and generation could be explored.

References

- Matt Barton and Shane Stacks. 2019. *Dungeons and desktops: The history of computer role-playing games 2e*. CRC Press.
- Nacir Bouali and Violetta Cavalli-Sforza. 2023. [A review of text-to-animation systems](#). *IEEE Access*, 11:86071–86087.
- Jonathan Cooper. 2021. *Game anim: video game animation explained*. Crc Press.
- Kaveh Hassani and Won-Sook Lee. 2016. Visualizing natural language descriptions: A survey. *ACM Computing Surveys (CSUR)*, 49(1):1–34.
- Thomas Hayton, Julie Porteous, Joao Ferreira, and Alan Lindsay. 2020. Narrative planning model acquisition from text summaries and descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1709–1716.
- J Hoberman. 1982. Disney animation: The illusion of life. *Film Comment*, 18(1):67.

- Ollie Johnston and Frank Thomas. 1981. *The Illusion of Life: Disney Animation*. Abbeville Press.
- Kate Kearns. 2017. *Semantics*. Bloomsbury Publishing.
- Nick Montfort and Ian Bogost. 2020. *Racing the beam: The Atari video computer system*. Mit Press.
- Masaki Oshita. 2010. Generating animation from natural language texts and semantic analysis for motion search and scheduling. *The Visual Computer*, 26:339–352.
- Lucia Santaella. 2009. *Mapa do jogo: a diversidade cultural dos games*. Cengage learning.
- Ari Shapiro. 2011. Building a character animation system. In *Motion in Games: 4th International Conference, MIG 2011, Edinburgh, UK, November 13-15, 2011. Proceedings 4*, pages 98–109. Springer.
- Lee Sheldon. 2022. *Character development and storytelling for games*. CRC Press.