Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool

Gulnara Sahibgareeva[1[0000-0003-4673-3253]], Oleg Bedrin[1[0000-0003-3300-4318]] and Vlada Kugurakova[1[0000-0002-1552-4910]]

[1] Kazan Federal University, Kremlin str., 18, 420111, Kazan, Russia

Abstract. This work belongs to the field of development and research of video games. In particular, the narrative part of video games is considered. The paper examines the existing practices of designing, developing and testing interactive storytelling of video games. The paper suggests a tool that automates these processes is proposed. The practice of developing a scenario prototype is described. Its effectiveness in development has been proven. The implications of the introduction of practice in the development of the narrative of video games are analyzed. The paper gives an analysis of relevant reference tools. Existing text-based visualization tools are analyzed. The paper forms the vision of the tool and shows its development progress. Describes the implementation of the storyboard visualization component. The paper proves the effectiveness of the development and sets plans for the future. In the future, work will continue to develop and expand the functionality of the script prototype generation tool.

Keywords: Video Game Development, Interactive Storytelling, Scenario Prototype, Narrative Design, Screenwriting, Game Documentation, Storyboard Generation, Interactive Storyboard.

1 Introduction

Video game development is a long and expensive process. Any optimization and automation can play a key role in the success of the final product.

The complexity of developing video games is in its composition. It includes visualization, audio, interactive interaction, game rules and story. Each component is closely related to the others, and together they give a unique result for any media – the experience of interactive consumption of content with a dramatic background. In other words, the players make their own story. The game development industry is a young and rapidly developing area. Specifically, the narrative component has long remained a burden rather than a powerful tool. Only in recent years its own theory and its own experts in this direction has begun to appear.

This paper provides an analysis of existing practices for developing interactive storytelling, as well as the results of attempts to automate this process.

The global goal of this paper is to introduce the development process of a tool that is designed to automate the generation of a video game narrative prototype, and also to capture the author's tools, such as a scenario prototype and interactive storyboard.

2 Formats of Technical Documentation of the Game Writer

The game writer creates the scenario in the form of technical documentation. There are several formats in which game writers can present the story of the game to the customer and the development team: text document, table, wiki markup, paper prototype, storyboard.

2.1 Text Document

The script of the main plot of the game, side branches and cut scenes can be presented in the format of a text document, as well as texts such as encyclopedias, letters, and notes found in the game. However, the text requires a lot of time and effort to explore.

The essence of the screenwriter's work is to accurately convey the story first to the members of the development team and all those who make important decisions about the game, and then to the player. Without understanding the story that the game should deliver, it is impossible to create a harmonious connection between its components.

The effectiveness of text perception is relatively low, in contrast to video and games. It has been proven that the efficiency of visualization [1] and, especially, immersion [2] is significantly higher than the efficiency of information transfer through text.

After the work of the screenwriter has been examined, edits are made to it. Any edit can lead to an avalanche effect of changes in the script. It is difficult to eliminate the consequences of this effect if the scenario is not visualized at least in the form of a graph structure. When creating games, more efficient methods of creating, storing, and transmitting documentation are needed. Best of all, the text is enhanced by visualization, as well as the ability to immerse yourself in the narrative.

2.2 Table

Most of the content that will be further developed by the specialists can be presented in tabular form. Let's analyze the principle of operation of tables on the example of an imaginary quest. (see Table 1).

Table 1. An example of a tabulation for the quest

№	Quest name	Character	Character replica	First step	Second step	Third step	Fourth step
1	Birthday	Ann	Today is	Buy	Choose a	Bake the	Present a

cake	Katya's birthday. Bake her a cake.	products.	recipe.	cakes.	cake.

In this example, we see one of the possible representations of the quest for a mobile game. For each quest, the following parameters are highlighted in the example: identification number, name, name of the character who will give the quest to the player, his replica for giving the quest, the stages of the quest implementation by the player's actions.

Let's analyze the need for these elements:

- the quest number is required to speed up communication between development team members;
- the name of the quest is also necessary for communication, but it is also displayed in the game itself;
- the player must somehow get the quest. The quest is given by a specific character, who utters a certain remark for this purpose;
- to finish the quest, the player must complete a certain amount of actions.

The number of parameters and rows in the table can be expanded to the amount that will be needed in development. Tabular form is more effective than text document, because it visually presents information in a structured form.

Tabular form is not only suitable for quests. It can be used for any task, specifying the necessary texts, animations, and sound can also be done with it.

Table is a good technical tool designed to structure information and organize the work of a team. However, it is not suitable for the presentation of the game's plot and subsequent testing and evaluation, since the table has nothing to do with the gameplay.

2.3 Wiki Markup

The hierarchy supported by wiki markup allows you to create a whole system of documents that will store a holistic view of the in-game world. How wiki markup works: each text document contains concepts (words, phrases), the essence of which is revealed in other documents, which can be accessed by a link from the current document.

The advantage of wiki markup is that it allows you to maintain lore in an ordered, systematized form.

However, these are still the same text documents that are split into a series of web pages. The development team and everyone involved will have to spend time studying all the information.

2.4 Paper Prototype

A paper prototype is a great solution for both prototyping game mechanics and prototyping game storytelling [3]. If the game writer was able to guide the audience and

participants through the story with the help of improvised means, and everyone liked the story, then, theoretically, the story might be liked in the final form of the game. Speaking of paper prototyping of the story, it is important to note that this is the same game, but devoid of implemented rules and mechanics, simplified so that participants plunge only into a fictional world, but do not delve into the game process [4].

On the one hand, this is not good in terms of narrative design. On the other hand, narrative design itself has a number of evaluation tools that allows testing the narrative for specific parameters. These parameters are immersion, tempo-rhythm (pacing), empathy.

Immersion [5] is a state where the player forgets about time and reality, dives into the plot of the game, realizes its importance for himself. Immersion is a familiar state where we can find ourselves when we read books and watch movies. It is a feeling from which we return as from a dream.

Tempo-rhythm [6] is a frequency and density of plot representation. The player must become familiar with the world of the game gradually. This is important for two reasons: first, getting to know something new should happen by stages. And second, there must be an intrigue that will keep the player in the process of learning the story. Tempo-rhythm is regulated exclusively through testing, and a paper prototype allows this to be done.

Empathy [7] is an important feeling that a player has in relation to the characters of the game. Familiar characters the player can understand and sympathize with cause empathy. Everyone can find something for themselves in a character. But there are also failed characters that are either very weak themselves, or poorly invented, or they do not meet the expectations of the target audience, or the current time. It can also be checked during testing.

Therefore, a paper prototype is an effective and cheap tool for testing game plot at an early stage [8]. To create a paper prototype, you need the presence of the author or instructions for conducting a testing session, as well as minimal in-game artifacts created with the help of improvised means, for example, items necessary for the plot – a sword, coins, and books.

2.5 Storyboard

One way to visualize a scenario is a storyboard. A storyboard is a series of sketches of a future game. It can be used for creating movies, cartoons, advertising, and any audiovisual work which requires adjusting the position of the camera and the placement of objects.

In our concept, the storyboard of the game scenario can be called an interactive storyboard. An interactive storyboard is a series of images that contain a visualization of the scene, required characters, their position, their names and replicas. In fact, these are snapshots of what the player will see during the game. It is very similar to the existing game genre – visual novel.

Visual novel is a game genre in which you can make a choice of actions and replicas in dialogues, it is a set of backgrounds, pictures of characters and text. Hence, the

game scenario presented in the form of an interactive storyboard is already a game, a minimal product that can be presented to everyone involved, and get feedback.

In addition to visualization, an interactive storyboard, like a visual novel, can include text, sounds and music, and simple animations.

Interactive storyboard is one of the best ways to present the game's narrative. Visualization helps to present the complete atmosphere of the game in the form of a series of sketches, or collected from pictures and photos from the internet, even if it is rough. It will take time to develop a storyboard, but it will be much more effective than all other ways of presenting the story described above. In addition, the storyboard can be shown to everyone involved without the personal presence of the screenwriter.

3 Scenario Prototype

A scenario prototype is an interactive prototype of the game's story without game mechanics.

An interactive storyboard is a variation of a scenario prototype.

There are several recommendations for creating a narrative prototype: it should match the genre of the game being developed, have stubs on the gameplay and game mechanics, and be developed on the target engine.

Matching the scenario prototype to the genre and developing it on the target engine are recommendations that eliminate possible difficulties with applying the scenario prototype to the game. In addition, it is easier to increase the functionality of a scenario prototype than to develop a game from scratch. Having a ready-made scenario prototype for some games means having one of the development stages. Thus, the scenario prototype reduces the amount of work during the development phase.

Gameplay and mechanics stubs assume that a scenario prototype is created to test hypotheses and test the quality of the game's narrative, but not its gaming component. Separating the gameplay from the story is not a good thing, but is possible and, in our opinion, effective at some point. Raw gameplay can be the reason why the audience dislikes the story. Therefore, without gameplay, it is easier to look into the game plot. A scenario prototype is an attempt to temporarily divide the story and the game in order to test it for immersion, tempo-rhythm, and empathy.

A scenario prototype allows developers to save project resources and test the story before starting the development of the content of the game.

The scenario prototype is applicable for both small and large projects, both for development from scratch, and for completion of the existing game.

A scenario prototype is based on the work of the screenwriter in the form of technical documentation and is developed by a narrative designer, who is a specialist in the implementation of story in games.

4 Generating a Scenario Prototype

Generating a scenario prototype instead of manual assembling will help solving two problems: relieving the narrative designer and reducing the threshold for game writers to enter the industry.

The first problem is that at the preliminary stage of development, the narrative designer is not only engaged in assembling a scenario prototype. The narrative designer also needs to create a concept for introducing storytelling into the game. When this one task is removed from the narrative designer, he will be able to focus on other things.

The other problem is also about easing other specialists' load. If there is a scenario prototype, the game writer is able to find some inconsistencies in the script and correct them on their own. Then the game writer will have some time to study the game engine and programming, which is important when choosing personnel in the gaming industry.

Thus, we formulate the value of our tool. The scenario prototype generator can save all possible resources that are available on a game development project. Moreover, we can see the application of our tool not only in the gaming industry, but also in the development industry of serious games and simulators.

5 Existing Solutions

Now there are many solutions for generating visual content based on natural language, which differ in goals and approaches. For example, in the 2016 study [9], 26 implemented tools are reviewed. The tools in this paper are divided into several types: converting text into a series of images (Story Picturing Engine), converting text into a series of associated images (Text-to-Picture Synthesis System), converting text into a set of static scenes (TEXT-TO-SCENE), converting text into animation, including the position of objects (TEXT-TO-ANIMATION).

The accuracy of visualization generation depends on the degree of development of NLP and neural networks, as well as the computing powers. However, there are already acceptable generation results that can be used in the cinema industry [10] and animation [11], [12].

We highlighted some of the visualization tools in this study and intend to adapt it for our own development.

ScriptViz [13] allows visualizing the script as a three-dimensional scene, the content of which is generated in real time. The user enters the script, taking into account the following restrictions: the text must be written in English, have clear wording and a good structure.

SceneMaker [14] is a closed development, the purpose of which was to automatically generate a number of artifacts for film production. It was planned to generate scenes from the script exactly to the facial expressions of virtual actors, lighting and audio accompaniment.

CRAFT is a neural network that is able to choose the appropriate animation material based on the descriptive text. Based on 25184 fragments from the animated series The Flintstones, the neural network compiles new episodes of the cartoon by the text description. There are still errors in compiled episodes like incorrect overlay of objects or incorrect selection of animations.

These tools support our hypothesis that the technical documentation of game writers can be visualized programmatically.

However, prototyping the narrative of games differs from tools for cinematography and animation mentioned above that it implements the possibilities of interactive interaction, and the variability of the plot.

We have found tools that are also applicable to games.

StoryFlow [15]. When we think about visualizing the structure of a story, the idea of presenting information in the form of a graph comes to mind. But the creators of the StoryFlow tool went further and presented the structure in the form of a special Yarn structure. It is able to visualize the variability in the events that occur, which is extremely useful for a beautiful and clear visualization of game events, character interaction and determining time intervals.

Machination Tool [16] is a web-based platform for designing, balancing and modeling game systems. It allows you to visualize and simulate game systems in the form of balance charts. We see this tool as one of the elements of a game prototype rather than a scenario one.

Another one tool is interesting in terms of visualization capabilities and as an aggregator of various software solutions.

Orange3 [17] is a visual programming software package based on components for data visualization, machine learning, and data analysis. The program includes hundreds of ready-made nodes: various visualizers (diagrams, graphs, tables), algorithms for processing and preprocessing texts, neural networks that are ready for training and operation, and each node is responsible for its part of the work. In addition, developers provide users with the opportunity to implement their own functionality in the form of Python programs, for which there is a separate node.

The tool that inspired our development is Storyboarder.

Storyboarder is a tool for creating storyboards from Wonder Unit. This tool can generate shots based on entered text queries. However, it should be noted that (so far) Storyboarder understands only a limited amount of words and some alternatives for words from this list, which are placed in the ‘alternativeValue’ document.

6 Authors’ Vision of the Tool

Despite the fact that a comprehensive study has been done to analyze approaches to generating visualization, we have not found any suggestions for generating full-fledged scenario prototypes of games.

Our main goal is to develop a tool for generating a scenario prototype of the game that is based on the technical documentation of the game writer.

The process of working with the narrative element of the game using our tool, as follows:

1. writing a game script in the form of plain text in natural language in the form of technical documentation (text, tables, wiki markup);
2. generating a scenario prototype basing on documentation;
3. presenting and discussing the scenario prototype within the team, with the customer and testing on a focus group;
4. editing the scenario prototype.

Development plan for this tool:

1. to visualize the structure of a game scenario presented as a natural language text (further referred to as TEXT) in the form of graphs or Yarn representations;
2. to automatically assemble an interactive storyboard (for a limited number of genres, in the form of a visual novel or point-and-click game) of the game script from the text;
3. to create an algorithm to generate an interactive scenario prototype from text using the Unity game engine;
4. to automatically generate a Machination gameplay balance chart based on text;
5. to automatically assemble a game prototype of the game from the text.

We see it all in form of a complete application, which receives a series of text documents as input, and the output is a project that combines generated scenarios and prototypes of the game.

In the picture (see Fig. 1) there is the diagram of the tool that we have developed for subsequent implementation.

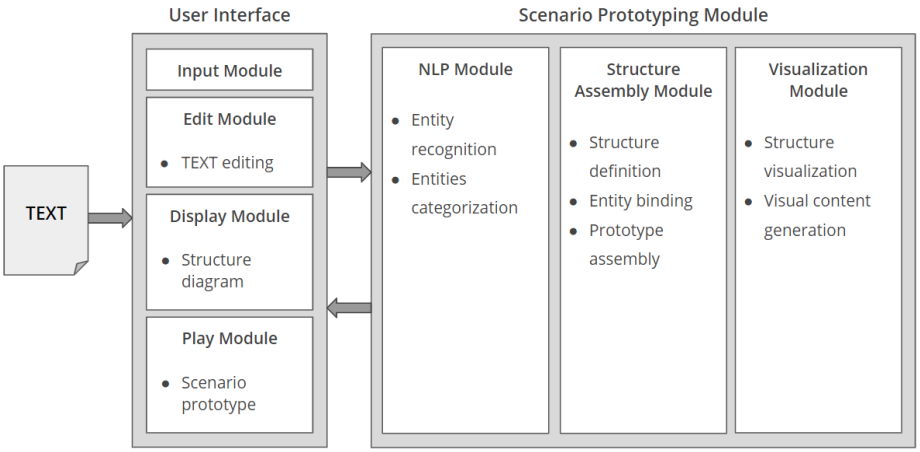

Fig. 1. Diagram of the tool operation

The input is a script (TEXT), which is written in natural language. We assume that it should follow some formatting rules for better object identification.

Edit Module allows the user to edit a text document.

In the Display Module mode, the user is introduced to the result of the structure visualization.

In Play Module mode, the user launches a scenario prototype that can be “played”.

The module for generating a scenario prototype includes three parts, each of which sequentially processes the result of the previous modules' work:

1. the NLP module recognizes the necessary objects and relationships between them;
2. the structure assembly module establishes a complete scenario structure, connects objects to each other, and builds a scenario prototype structure;
3. the visualization module selects the necessary three-dimensional models and distributes them according to the structure of the scenario prototype. Next, the module sends the resulting visualization of the scenario prototype and the visualized structure of the scenario, for example, in the form of Yarn, to the Play Module.

7 Work Accomplished

A whole “series” of our developments has already been focused on creating a generator of scenarios and game prototypes. In our paper [18], we described our own approach to creating a scenario prototype generator and a pilot implementation of a tool for creating scenes basing on objects extracted from the text. In another paper [19] we presented (see Fig. 2) a program for Orange 3, which takes a formalized text as input, and at the output shows a different visualization of the structure: the geometric connectivity of locations, in which locations the character appears, what cues are spoken in these scenes, etc.

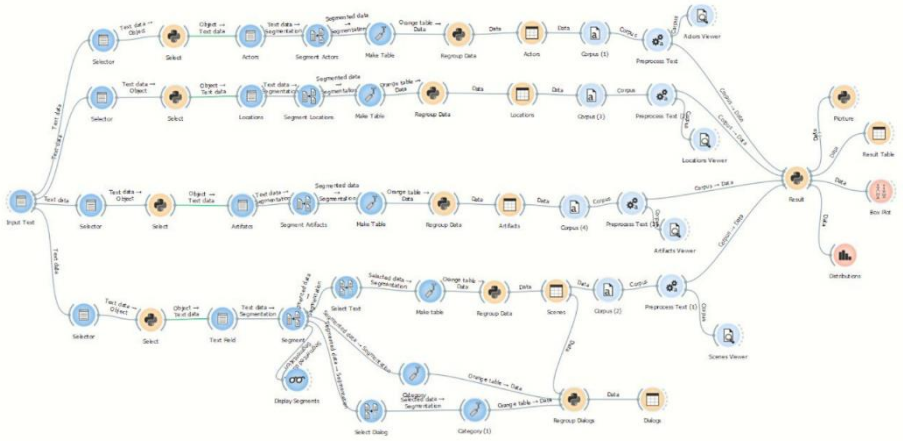

Fig. 2. Node-based scenario visualization generation system

Another work [20] is related to the solution of the problem of extracting information about a scene from the text. To do this, a text analysis system has been created. The necessary camera settings are calculated from the context. The work is at an

early stage: the functionality of text recognition of a shot, similar to Storyboarder, is being recreated (see Fig. 3).

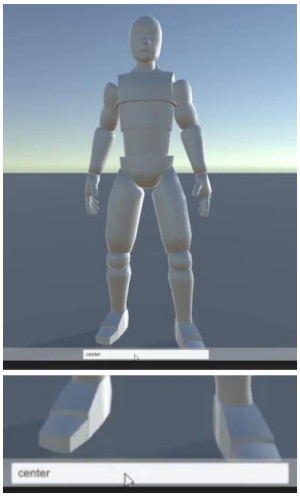

Fig. 3. Extracting information about a shot from text

8 Current Status of Work

One of the tasks of the scenario prototype generator visualization module is to generate an events visualization in space. This requires information about the characters in the scene, their actions and cues.

As part of the events visualization in space, characters should be positioned in poses that match the context. As a result, there should be a series of shots, which is a storyboard.

This is necessary as one of the stages of developing an interactive storyboard, in which the transition to the next shot will depend on the player's choice.

The Storyboarder tool was chosen as a platform for the experiment. It contains the necessary functionality for visualization generation.

The input is a text in English, the sentences with descriptions of separate scenes. Each sentence contains information about who is present in the scene and what they are doing. The text is analyzed and the character and its action are extracted from it for visualization. The result is processed and sent to the Storyboarder input, where shots are generated, and then shots are collected into a storyboard to evaluate the result.

The details of the technical implementation and the results of the experiment are shown below.

9 Technical Implementation of the Current Stage of Work

The process of generating an event in space by text input is an automatic pipeline. It is the reception of text and its processing. The result is then sent to Storyboarder for storyboard generation.

The ability to generate storyboard elements based on a given word vector is only present in the abbf5f24 commit cast of the open official storyboarder repository [21]. For this reason, this version and not the latest version is used in the experiment.

In order to integrate Storyboarder into the pipeline, we isolated the functionality that can call the generation procedure with an argument in the form of text and initiate all the necessary mechanisms for obtaining a storyboard. The input is a text that is managed by another software component, and the output is a generated storyboard.

We also implemented the Express micro web server. It accepts a request with a parameter in the form of input text on the loopback interface, passes this parameter to the isolated functionality and, as a response, sends the text that it received, so that the sender can verify the success of the generation operation.

The source text is converted into a body [22]. It is a set of texts selected and processed according to certain rules, used as a base for language research. The body is used for statistical analysis and testing of statistical hypotheses, confirming linguistic rules in a given language.

Since machine learning mostly deals with any functions where the image is a number or a set of numbers, the text needs to be vectorized in order to process it. This process is converting numbers into vectors according to certain rules, so that the loss of information contained in the text is minimal.

Also, one of the stages of text processing is the selection of stop words. These are the words that play a small role in the study of the properties of language. For example, the words “a”, “am”, “an”, “is”, “are”, etc.

The input text must be processed in order for Storyboarder to correctly generate the storyboard. The first step is tokenization. It is the conversion of text into a body in the form of an array of sentences, also each sentence must be split by words and punctuation should be removed. The second step is filtering all words in the body, which will clear it from stop words. The next is recognizing names and converting them into an impersonal female or male entity. It is crucial because Storyboarder can't distinguish names, but it can distinguish a gender of entities and then convert it to male or female models in a shot.

During the filtering stage, stop words collected in the appendix to the paper were used [23]. The filtering process is iterative, the words are compared character by character. At the next stage of the algorithm, it is necessary to build a classifier that will recognize names in the text and determine its gender.

A multiclass naive Bayes classifier was chosen as a model for a development, since names are given independently of each other. To train the classifier, the names from public data of the US Social Security Service are taken [24]. Each name in the training sample vectorizes into symbolic bigrams. There is also a frequency analysis, based on which the percentage of use of the name as a female and male is visible. The time spent on the process is minimal, since training a naive Bayes classifier is the

calculation of independent probabilities, and the product of it is in the denominator of the formula according to Bayes' theorem [25].

The trained Bayes classifier responds with ‘zero’ if the name that came to the input is female and ‘one’ if it is male. Using the classification operation, each time a name appears in the text, it is replaced with the word ‘man’ or ‘woman’, depending on the gender assigned to it by the trained classifier. Depersonalized words are selected as the most understandable for the Storyboarder sentence parser.

10 The Result

It should be noted that the natural language text was written while Storyboarder understands a limited list of formal words. It was important for our work to have a character to appear in the scene with the possible Storyboarder animation.

The camera position is set manually by a sequence of parameters that Storyboarder understands.

So, at the input, we submit the following text:

Bob is walking.

Alice says hi.

Bob is walking and looking back.

Bob is walking.

Alice hangs one arm.

Alice crossing hands.

Then, depending on the context of the sentence, we form the necessary camera settings. The settings for each offer are stored in a separate file:

1. looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside;
2. looking forward, medium long, single person, centered, right angle, eye level, long lens, light, frontrightlit, outside.

List of conversion processes that are performed on the text:

tokenization with elimination of punctuation and stop words;

recognition and depersonalization of names by replacing them with ‘man’/’woman’ depending on the gender of the name using a naive Bayesian classifier.

The resulting text is supplemented with camera settings using string concatenation.

This is how the result is generated, which is submitted to the input to the Storyboarder:

1. 'man walking, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';
2. 'woman say hi, looking forward, medium long, single person, centered, right angle, eye level, long lens, light, frontrightlit, outside';
3. 'man walk look back, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';

4. 'man walking, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';
5. 'woman hang one arm, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside';
6. 'woman crossing hands, looking forward, medium long, single person, centered, left angle, eye level, long lens, light, frontrightlit, outside'.

Storyboarder automatically generates storyboard shots based on the received data. In order for the shots to become part of the storyboard, they need to be added to the storyline manually (see Fig. 4).

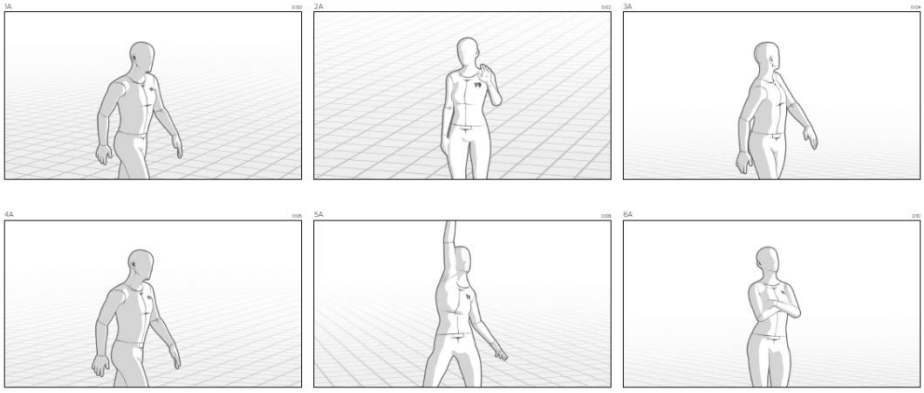

Fig. 4. The resulting storyboard

As can be seen, it is possible to use Storyboarder to generate a comic only when the text is prepared. Expanding the functionality and body of Storyboarder will allow us to work more effectively with the visualization of natural language text. We see further development in the framework of our own application.

11 Future Plans

Some of our future plans and tasks that were shaped during the discussion of the development results:

1. to extract information about the shot from the text;
2. to create a layout and to form technical documentation;
3. to create an interactive storyboard;
4. to generate a visualization.

Let’s describe each task in more detail.

Setting a shot based on the context of the story remains unrealized. This is important in terms of automating the generation of scenario visualization, and we have already started development in this direction.

We did not concern the classic markup of scenario documents for non-interactive works and the specificities of scenarios for interactive works. We plan on exploring it

in our future papers. Once the issue of markup and formatting of documents with a non-linear narrative is solved, this will help working on the following task related to the interactivity of the storyboard.

Storyboards are good in cinematography and animation, as well as in games, for example, in creating cutscenes. We propose adapting the use of storyboards to develop a game narrative. We called it interactive storyboarding. It implies that the sequence of shots depends on the choice of a potential player. The implementation of this perspective is one of our next tasks. Interactive storyboard is a bit like a visual novel or a point-and-click game. In the game, the narrative moves at the expense of the player's actions. Thus, the next task we set is to bring the storyboard to an interactive form.

12 Conclusion

The paper introduces the term scenario prototype and describes the authors' idea of it. Moreover, it forms the relevance of the scenario prototype generation tool.

The paper provides an analysis of the tools that inspired the authors. The existence of these tools suggests that the development of a scenario prototype generation tool is possible.

This is confirmed by examples of our colleagues' tools that were developed with the participation of the authors of this paper.

As an experiment, a prototype of the visualization component of the scenario prototype generation tool was developed. The prototype generates a storyboard based on text. It is designed to save time on visualizing the game narrative.

As a result, the requirements for the generation tool were detailed. Discourse about the development of the tool led us to establish a number of tasks for the next stages of prototyping.

This development is interesting in terms of creating games, serious games, simulators, which are also the subject of authors' close attention (see for example, [26], [27]).

References

1. Davies, D., Bathurst, D., Bathurst, R.: The Telling Image: The Changing Balance between Pictures and Words in a Technological Age. 1st edn. Clarendon Press, Oxford (1990).
2. Raja, D., Bowman, D., Lucas, J., North, C.: Exploring the benefits of immersion in abstract information visualization. In: 8th Int'l Immersive Projection Technology Workshop. Tech, Virginia (2004).
3. Koenitz, H., Dubbelman, T., Knoller, N., Roth, C., Haahr, M., Sezen, D., Sezen, T.: Card-Based Methods in Interactive Narrative Prototyping. In: International Conference on Interactive Digital Storytelling, pp. 552–555. Springer, Switzerland (2018).
4. Paper Tales: A Guide to Narrative Prototyping, https://www.gdcvault.com/play/1020509/Paper-Tales-A-Guide-to, last accessed 2020/12/17.

5. Cairns, P., Cox, A., Nordin, I.: Immersion in Digital Games: Review of Gaming Experience Research. Handbook of Digital Games, 337–361 (2014).
6. Newman, J.: Videogames. Psychology Press, Routledge (2004).
7. Blanchard, L.: Creating empathy in video games. The University of Dublin, Dublin (2016).
8. Koivisto, E., Suomela, R.: Using prototypes in early pervasive game development. In: Sandbox07: An ACM Video Game Symposium, pp. 149–156. Association for Computing Machinery, New York (2007).
9. Hassani, K., Lee, W.-S.: Visualizing Natural Language Descriptions: A Survey. ACM Computing Surveys 49(1), (2016).
10. RivetAI, https://www.rivetai.com/, last accessed 2020/12/17.
11. Gupta, T., Schwenk, D., Farhadi, A., Hoiem, D., Kembhavi, A.: Imagine This! Scripts to Compositions to Videos. Cornell University, Cornell (2018).
12. Storyboarder, https://wonderunit.com/storyboarder/, last accessed 2020/12/17.
13. Liu, Z.-Q., Leung, K.-M.: Script visualization (ScriptViz): A smart system that makes writing fun. Soft Computing 10(1), pp. 34–40 (2006).
14. Akser, M., Bridges, B., Campo, G., Cheddad, A., Curran, K., Fitzpatrick, L., Hamilton, L., Harding, J., Leath, T., Lunney, T., Lyons, F., Ma, M., Macrae, J., Maguire, T., McCaughey, A., McClory, E., McCollum, V., Mc Kevitt, P., Melvin, A., Moore, P., Mulholland, E., Muñoz, K., O'Hanlon, G., Roman, L.: SceneMaker: Creative technology for digital storytelling. Springer, Switzerland (2016).
15. Padia, K., Bandara, K., Healey, C.: A system for generating storyline visualizations using hierarchical task network planning. Computers & Graphics, 78, 64–75 (2019).
16. The Designer's Notebook: Machinations, A New Way to Design Game Mechanics, https://www.gamasutra.com/view/feature/176033/the_designers_notebookc, last accessed 2020/12/17.
17. Orange 3, https://orange.biolab.si/, last accessed 2020/12/17.
18. Sahibgareeva, G., Kugurakova, V.: The concept of automatic creation tool for computer game scenario prototype. Russian Digital Libraries Journal, 21 (3-4), pp. 235–249 (2018).
19. Dobrokvashina, A., Gazizova, Je.: Automatization of a gaming prototype development based on the result of processing of a formalized game design document. Uchenye zapiski ISGZ, 17 (1), pp. 583–589 (2019).
20. Astaf'ev, A.: Development of a tool for assembling scenes by tags. Kazan Federal University (2019).
21. Storyboarder GitHub, https://github.com/wonderunit/storyboarder, last accessed 2020/12/17.
22. Nikolaev, I.S., Mitrenina, O.V., Lando, T.M.: Prikladnaja i komp'juternaja lingvistika. M. URSS (2016).
23. Nothman, J., Qin, H., Yurchak, R.: Stop Word Lists in Free Open-source Software Packages. In: Workshop for NLP Open Source Software. Association for Computational Linguistics (2018).
24. Baby Names from Social Security Card Applications, https://catalog.data.gov/dataset/baby-names-from-social-security-card-applications-national-level-data, last accessed 2020/12/17.
25. McCreery, C.: First-year Statistics for Psychology Students Through Worked Examples. 1. Probability and Bayes' Theorem. Oxford Forum (2018).
26. Abramov, V., Kugurakova, V., Rizvanov, A., Abramskiy, M., Manakhov, N., Evstafiev, M., Ivanov, D.: Virtual Biotechnological Lab Development. BioNanoScience 7(2), pp. 363–365. (2018).

27. Kugurakova, V., Abramov, V., Sultanova, R., Tsivilskiy, I., Talanov, M.: Virtual Reality-Based Immersive Simulation for Invasive Surgery Training. European Journal of Clinical Investigation, 48, 224–225 (2018).