V-SET 2: Visual Story Editing Tool V2

Subtitle: The Interactive Narrative Midterm Project Write Up.

Student ID: 200110436

Unity ID: ychen74

Name: Yi-Chun Chen

Revision History:

Because the final project is an expansion for midterm project, the updated parts are marked in revision history.

Tool Version	Update
V-SET v1.0.0 (Midterm project)	Introduction, System description, Evaluation with cogtool.
V-SET v2.0.0 (Final project, new start from P4)	Class diagram, Function details, Interface

Outline

Introduction: Related Issues and Proposed Tool
System Description
System Design: class diagram (new)
Functions: implementation details (new)
Interface (new)
Unimplemented functions (new)
Demo Video & Github Link
Reference
License Information and Permission Information of Sample Artworks

Introduction: Related Issues and Proposed Tool

In current story engines, there are some authoring issues beyond tools [1]. One of the issues is the lack of useful graphical user interface, for authoring tools, which may allow authors to edit story content without working on programming language/ scripts. It causes the blur between story world and the engines. And the lack of useful graphical user interface also make authors hard to observe the generated story from different perspectives. Another related issue is that current story engines usually generate the story world as 3D simulations. Even though 3D environments are more realistic, they cost more time and computational resources to display the generated story content. As a result, authors cannot quickly see the results of generated content.

The Visual Story Editing Tool (V-SET) is an attempt to try to tackle part of these issues; it is a prototype system for graphical user interface authoring tool and will quickly display the story content through 2D graphic instead of 3D models. V-SET divides story as events which scatter on characters' storylines, and then allows authors to arrange the events' time orders as well as to edit the elements inside story events through graphical interface hence to change the story content. The tool uses built in display queue and display story events chronologically by adding story events into queue.

This report is going to introduce this tool. And the following content are organized in this order: the next section will describe the brief structure of this system, the third section will show the detail design (class diagram) of the system. Then, the next part will further introduce functions of the system in detail. In the final part, this section will introduce current user interface and unimplemented functions.

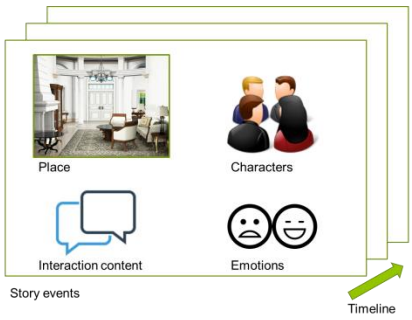
Brief System Description

This section will introduce the underlying idea of V-SET about how it decomposes a story into sub units, the data structures for representing a story, and the system structure of this tool.

Story decomposition

To find the small unit that composes a story and is suitable to be edited through graphical user interface, V-SET decomposes a story as a sequence of story events; and then defines events as a group of interactions between characters which happen in a short period of time and in a same place. Besides, considering that dialogue is one type of interactions which relatively easier to be displayed and usually be used to form story content, V-SET describe a story like the structure shown in Figure 1.

Fig. 1: A story is a sequence of story event, and an event contains a place, some characters, dialogues (interactions) between characters.



System Structure

To display a story and to be an authoring tool, below are functions the V-SET would like to support:

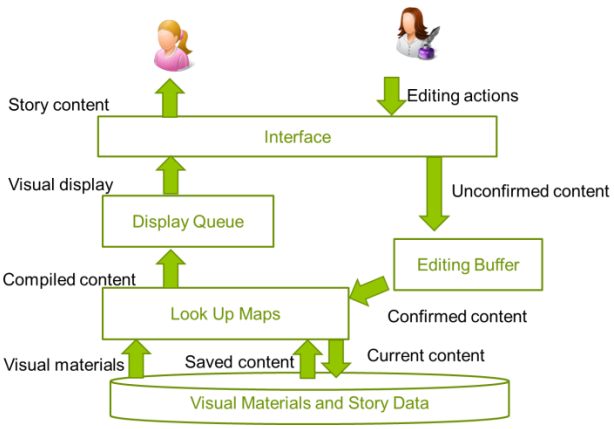
1. **Character customizing**: This function allows authors to customize characters' visual image through alternate the image of physical characteristics.
2. **Visualized storyline**: This function allows authors to directly see and arrange story events on storyline in arbitrary order, and provide a perspective of story structure.
3. **Event editing**: This function allows authors to modify the details of a story event: including modifying the event place, adding dialogue content and assign dialogue to specific characters.
4. **Visual result generating**: After authors finish editing storyline, the tool help to compile the story and generate 2D visual results. It will display the story events according to author's arrangement.

Therefore, to support these functions, the system structure of V-SET is divided as five sub parts. The first one is the *Interface*, which is used to interact with authors and readers. It take authors editing actions as input, and then will transform the inputs into runtime data structure and store the changes in the second part—the *Editing Buffer*. Not until the authors confirm the changes on story content, the changed content will cached in *Editing Buffer*. That is because the tool should support the flexibility of canceling current editing. Once the changes are confirmed by authors, these changes (such as the changes of character information, content of story, and so on) will update the recorded data in the third part: *Look up Maps.*

In *Look up Maps*, hash map are widely used to record the information of character, event, and visual materials, because all of these things are frequently accessed during the editing process. Hash map can provide better performance in such situations. The *Look up Maps* exchanges data with the fourth part—*Storage* while saving/ loading data and retrieving character features images. The fifth part of this system is the *Display Queue*. Once authors finish editing stories, they can

ask V-SET to compile current story at any time. The compile function will trigger the actions that adding current story events into *Display Queue* chronologically. Then, the Display Queue will combine character images, dialogue content together to show the generated story to authors or readers. Figure 2 shows the data flow and the structure of the system.

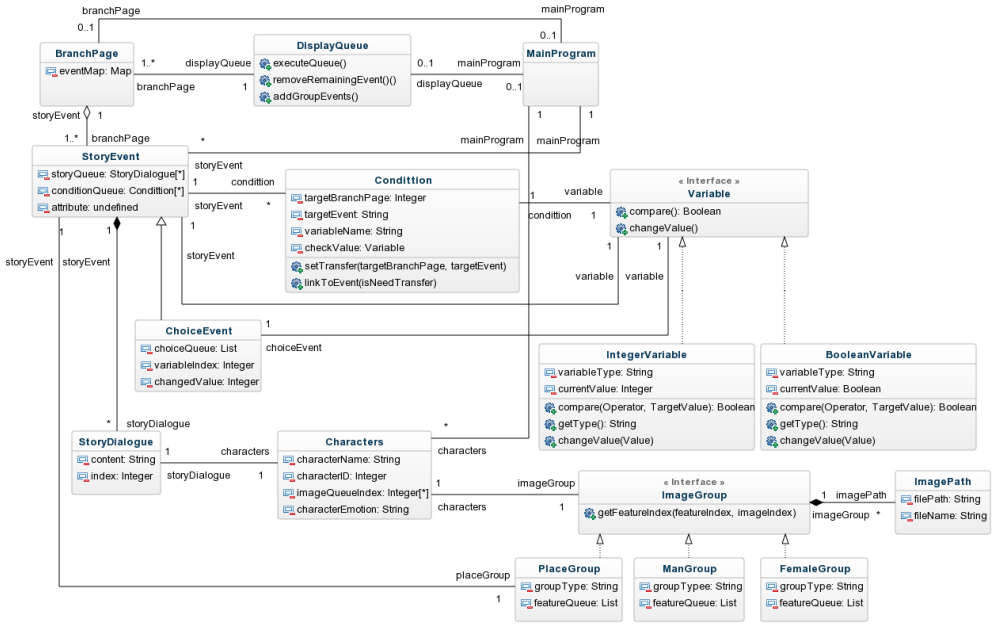
Fig. 2. The data flow of the whole process from editing story to generating result, and how data moves in-between system parts.



System Design: class diagram

This section will briefly introduce the detail design of the tool. Figure 3, shows the UML Class-Diagram of the tool.

Fig. 3.The class diagram of the tool



As shown in the figure above, the core of the tool is StoryEvent class. Besides as the descriptions in previous section, it contains character, place, and dialogue content. In each event, it also has three different queues, including dialogue queue, condition queue, and choice queue. The dialogue queue contains all of the dialogue related to an event. Condition queue records transitions between story events, whenever an event is successfully displayed, the system will then check conditions in condition queue to see what should be displayed in next round.

The process of this tool can be described as following: Users first construct several characters by choosing the character image as well as name, and then create story events which incorporate all the characters. When constructing events, users can create multiple branches as different story lines. The different story line implies the different sequence as well as various ending of a story. After finishing the story branches, users can create multiple choice events, which related to some variables. Each answer of the multiple choice questions will lead to some change of variable (E.g. choose choice1, variable will add 5). Therefore, users can further set conditions in story event. Whenever a story event is displayed, it will check all of conditions to see whether next event is in different story branch or not.

After the user finishing construction of the story, the tool will add story events of default branch into display queue. And then, during display, whenever user's choice changes the variables, Conditions will be checked to see whether the variable changes influence the story sequence or not. If the story sequence is changed, conditions will work as transitions between events. It can decide which event should be displayed

Functions: implementation details

Besides the functions in brief system descriptions, here are some new functions in version 2:

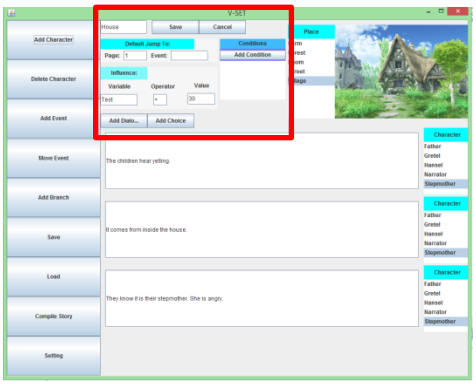
- **Add variables (parameters):** To parameterize the transition between story events, the tool provide the freedom of creating and changing variables for users. Each variable is divided as three parts; one is the variable itself, another one is the operators which can apply to the variables, and then the final part is the value of the variables.
- **Condition verification methods:** after display queue finish showing an event, all of related condition will be checked. This part use similar way with variable part, the variable are also check by being divided as three separate parts, which are variable, operator and values.
- **Multiple choices questions control variables:** In order to put the multiple choices questions into story process, the multiple choices inherit StoryEvent and using similar way with dialogue to create new questions. Creating the multiple choices, meanwhile, users can assign each choice a variable and change value of the variables, this step combines user input and transition condition together, thus users can use the choices to distinguish different story branches.
- **Modified display queue:** Different from previous queue. This queue become more dynamic, in the display process, when event conditions are met, the display queue should decide

whether it needs to re-compute or not. If story events move from one branch to another branch, the display queue need to find the right event and then add it in queue.

Interface

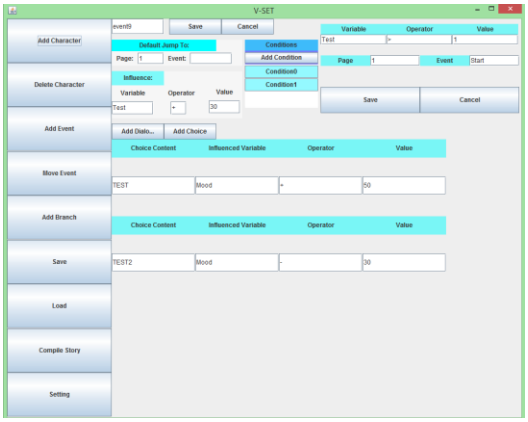
The figure 4 and 5 shows the interface of V-SET version 2. The description of each part will be presented below:

Fig. 4, add dialogue to story event.



The rectangle area is the editing panel for checks that should be evaluated after the story event is displayed. After this event, the influence will apply to assigned variables and all of conditions will be check. If there is no conditions matched, the event will choose the Default Jump as next event.

Fig. 5, add multiple choice to story event.



This figure shows that how to set multiple choices and also the setting of conditions. Once users decide the Boolean condition (E.g. if Mood > 50 = true)of transitions, they can set choices with corresponding variables. Then the result of users' choice can apply to transition conditions.

Unimplemented functions

Some unimplemented functions, because of time and difficulties, are listed in here.

- **Display narrative structure:** Originally, besides the story editing functions, this tool plan to provide different perspective of narrative structure. However, the structure is hard to be shown in 2D interface, especially when some story events are shared by multiple characters. For example, the triangle pairs of story events: event1 shared by(character1, character2), event2 shared by(character2, character3), and event3 shared by(character3, character1). It is hard to show this relationship in 2D interface.
- **Iterations of UI design:** Because this project aims to develop a user interface tool, multiple iterations of UI design are needed. But, I didn't spend a lot time to iteratively test and modify the interface design.
- **Error Handling:** Since the editing tool requires a lot of user input, for some input text fields, not all possible errors are handled. Therefore, the system might have some problem if users use incorrect input form.

Demo Video of Version 2

https://youtu.be/9XHCIR1AMVU

Github link

https://github.com/RimiChen/V_SET

Reference

[1] Spierling, U., and N. Szilas. "*Authoring Issues beyond Tools*." Interactive Storytelling, 2009, 50–61.

License and Permission Information of Sample Artworks

Backgrounds and Dialogue Frame

1. 誰そ彼亭 http://may.force.mepage.jp/

   License information (quoted from site): 有料無料、ジャンル、年齢制限の有無、などを問わず自主制作（同人）ゲームの背景、CG の背景、自サイトの壁紙などとしてご

自由にお使いいただけます。商用 (= 営利目的) 、企業さまのゲームにはお使いいただけません。同人ゲームでの有料頒布は OK です。

2. びたちー素材館 http://www.vita-chi.net/
   License information: http://www.vita-chi.net/sozai1.htm

Characters Artworks

Artist: Yi-Chun Chen (Rimi).