The *WideRuled* Story Generator Tutorial

Alex Mitchell
Communications and New Media Programme
National University of Singapore

Table of Contents

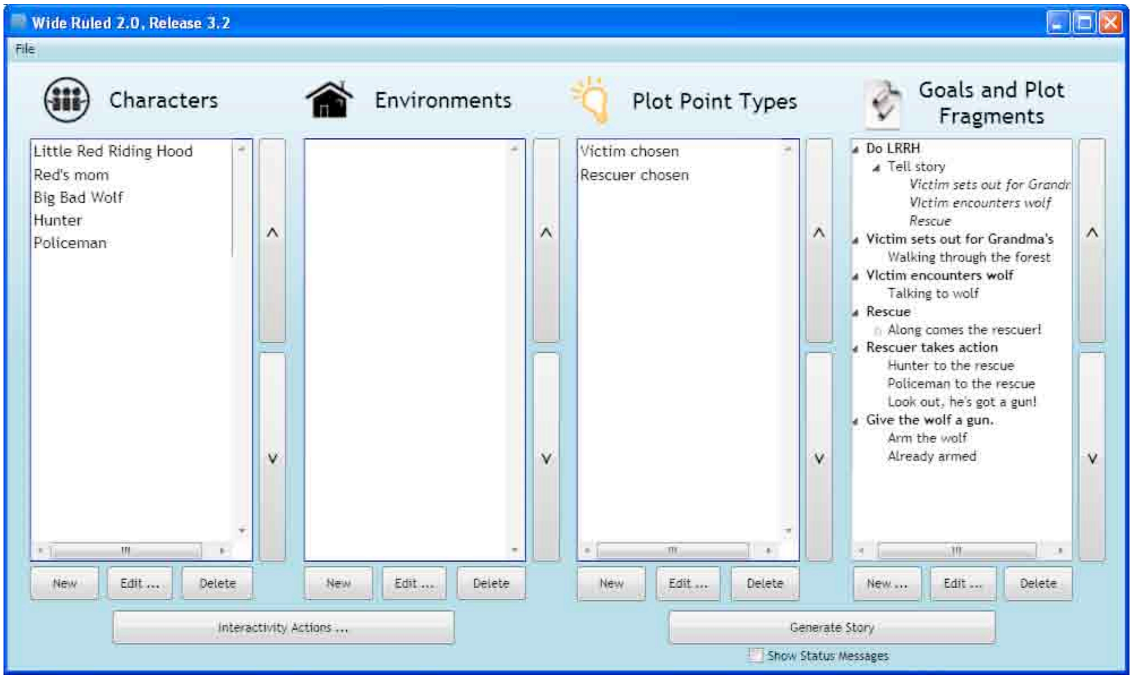
1. Introduction

In this tutorial, we will build a basic procedurally generated story in the **WideRuled** procedural story editor. Our example, "Little Red Riding Hood", tells one of several possible stories of a potential victim, either Red or her mother, traveling through the forest to deliver food to Red's grandmother. On the way, the potential victim encounters the Big Bad Wolf, and tells him where she'd going. The wolf rushes ahead and eats Grandmother, and then, when the victim arrives, eats her as well. At that point, one of two possible rescuers, either the Hunter or the Policeman, happens to be passing by. The hunter will try to kill the wolf, whereas the policeman will try to arrest him. Throughout the story, the reader has one choice available to her – should the wolf be armed? If the reader has chosen to arm the wolf, the rescue attempt fails. Otherwise, the rescue is successful, and they all live happily ever after.

The example demonstrates several aspects of **WideRuled**:

- Creation of characters: Red, Red's mom, the Big Bad Wolf, the Hunter, and the Policeman
- Creation of a series of Author Goals
- Creation of alternative Plot Fragments to satisfy an Author Goal
- Creation of Plot Points to keep track of important information in the story, and
- Creation of interactivity

The final version of our story as seen in the WideRuled editor is shown in Figure 1.



Little Red Riding Hood
Red's mom
Big Bad Wolf
Hunter
Policeman

Victim chosen
Rescuer chosen

- Do LRRH
  - Tell story
    - *Victim sets out for Grandr*
    - *Victim encounters wolf*
    - *Rescue*
- Victim sets out for Grandma's
  - Walking through the forest
- Victim encounters wolf
  - Talking to wolf
- Rescue
  - Along comes the rescuer!
- Rescuer takes action
  - Hunter to the rescue
  - Policeman to the rescue
  - Look out, he's got a gun!
- Give the wolf a gun.
  - Arm the wolf
  - Already armed

Figure 1: Final version of LRRH in the WideRuled editor.

The final text of one version of the story is as follows:

> *Once upon a time... Little Red Riding Hood was walking through the forest to deliver some food to Grandma. In a clearing in the forest, Little Red Riding Hood came across a wolf. The wolf asked where she was going, and she told him.*
> *Interaction!*
> *The wolf has a valid gun license, and always carries at least a small pistol, if not a semi-automatic. Rushing ahead to Grandma's house, the wolf quickly ate Grandma, and hid in her bed, hoping to trick Little Red Riding Hood into thinking he was Grandma. Upon reaching Grandma's house, Little Red Riding Hood was shocked to see that Grandma was actualy a wolf. Immediately, the wolf gobbled up Little Red Riding Hood. At that moment, the Hunter happened to be walking past. Unfortunately for the Hunter, the wolf had a gun, which he pulled out just in time, and shot the Hunter in the head. Grandma and Little Red Riding Hood never managed to get out of the wolf's stomach. *** The end. ****
> *Story Completed!*

In this version of the story, the victim was Little Red Riding Hood, the rescuer was the Hunter, and the reader chose to arm the wolf.

2. Getting Started

Begin by running the **WideRuled** procedural story editor, by double-clicking on the **WideRuled** icon in **Program Files->WideRuled**. This will start WideRuled with a new project.

The main window of the **WideRuled** editor is shown in Figure 2.

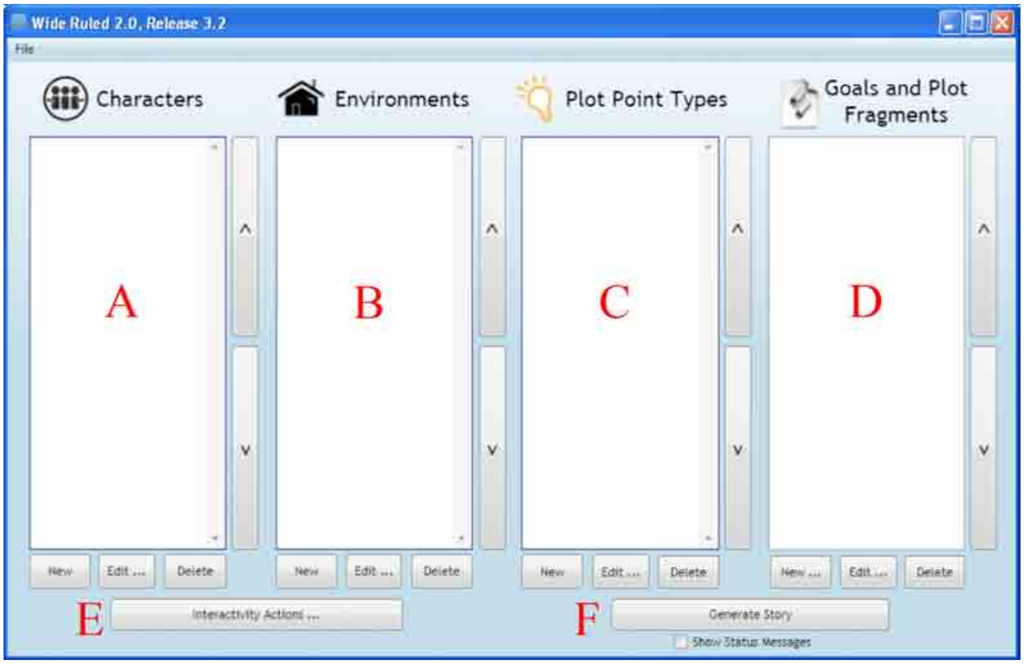


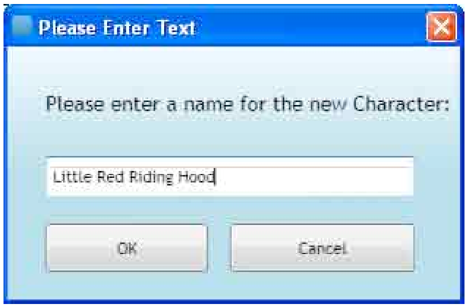
Figure 2: WideRuled main window.

The main window contains four lists: *Characters* (A), *Environments* (B), *Plot Points* (C), and *Goals and Plot Fragments* (D). Each contains a list of the relevant story elements, and allows you to create, edit and delete these story elements. There is also a button at the bottom of the main window for creating *Interactivity Actions* (E), and to *Generate* the story (F).

3. Creating Characters

First we will create the characters in the story, starting with Little Red Riding Hood.

1. Click on the *New* button at the bottom of the *Characters* list. A window titled *Please Enter Text* will appear (see Figure 3). Enter the name of the character, and press “Ok”.

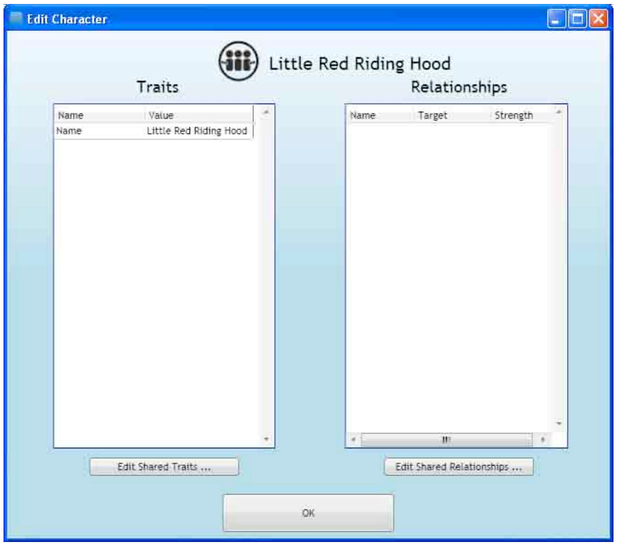


Please enter a name for the new Character:

Little Red Riding Hood

Figure 3: Creating a new character.

After entering the character’s name, you will see the *Edit Character* window (see Figure 4).



Name	Value
Name	Little Red Riding Hood

Name	Target	Strength

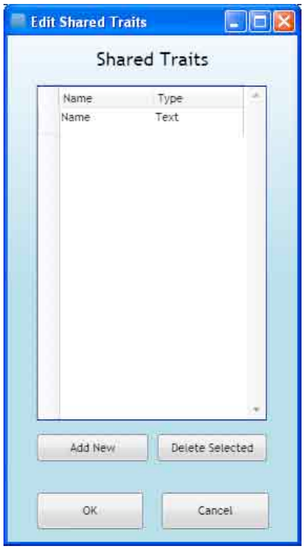
Figure 4: Edit character window.

The Edit Character window contains two lists, *Traits* and *Relationships*. All characters share the same type of traits and relationships, called *Shared Traits* and *Shared Relationships*. Individual characters can have different values for each of these traits and relationships. For example, all characters have a trait called *Name*, but each character has a different value for the *Name* trait.

There is always one shared trait, *Name*, which is used to identify a character. In addition, you can create a set of shared traits to describe the characters in your story. These will be used later, during story generation, to check for example if a character matches the prerequisites for an a plot fragment. For our story, we want our characters to have three additional shared traits: possible victim, possible rescuer, and armed.

2. Click on the "Edit Shared Traits…" button at the bottom of the *Traits* list. You will see the *Edit Shared Traits* window (see Figure 5). This list lets you add, edit and delete the list of traits that are available for all characters. Each trait has a Name, which identifies the trait, and a Type, which determines if the trait contains *Text*, *Numbers*, or *True/False*.

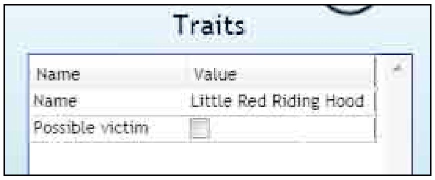
At this point, all we have is the default *Name* trait. We want to add three more traits.



Name	Type
Name	Text

Figure 5: Edit Shared Traits window

3. Click on the “Add New” button at the bottom of the list to create a new trait. A new trait will be added to the list, with Name “NewTrait”, and Type “Text”.
4. Select the text “NewTrait”, and edit it to change the name of our new trait to “Possible victim”. We will use this trait to identify which characters are possible victim’s of the wolf.
5. Next, select the text “Type” for our new trait. This is actually a pull-down menu. Select “True/False”. For each character, we will set this trait to “True” if the character is a potential victim.
6. Now click “Ok” at the bottom of the list. The window will close, and you will see your new trait in the Edit Character window (see Figure 6).



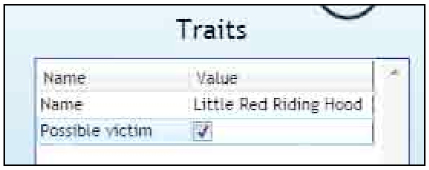
Traits

Name	Value
Name	Little Red Riding Hood
Possible victim	☐

Figure 6: New shared trait added to the characters.

Note that this trait is now available in **all** characters. If you add or delete a shared trait, this affects all characters in the story. The same applies to shared relationships.

7. To set a trait in a character, edit the “Value” column for the trait in the Traits list in the Edit Character window. For Little Red Riding Hood, we want her to be a possible victim, so check the check-box in the “Value” column for “Possible victim” (see Figure 7). This tells **WideRuled** that Red is a possible victim.

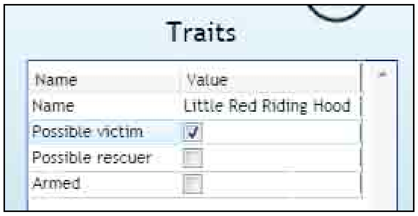


Traits

Name	Value
Name	Little Red Riding Hood
Possible victim	☑

Figure 7: Setting a trait in a character.

8. Now we will add the “possible rescuer” shared trait. In the Edit Character window, click on Edit Shared Traits…, and add a new trait. Name the trait “Possible rescuer”, and set its Type to “True/False”. This represents whether a character is a possible rescuer.
9. Add another shared trait, name it “Armed”, and also set its Type to “True/False”. This represents whether a character has a weapon.
10. Click on “Ok” to close the “Edit Shared Traits” window. The Edit Character window for Little Red Riding Hood should now show four traits, as shown in Figure 8. These are all the traits that we will need for our example.



Traits

Name	Value
Name	Little Red Riding Hood
Possible victim	☑
Possible rescuer	☐
Armed	☐

Figure 8: The complete set of traits for our example story.

11. Since Little Red Riding Hood is not a possible rescuer, and is not armed, leave these traits unchecked (False), and click on “Ok” to close the Edit Character window.

We’ve completed creating our first character, Little Red Riding Hood. You should see the character listed in the *Characters* list of the main window (see Figure 9).

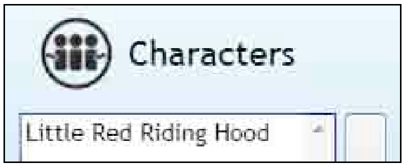
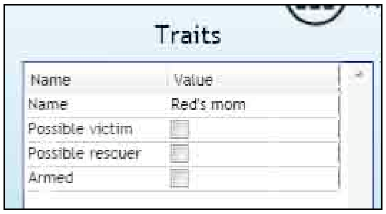


Figure 9: Our first character

Now we will create our remaining characters: Red's mom, the Big Bad Wolf, the Hunter, and the Policeman. Note that we don't need to create Grandma as a character, since she doesn't actually do anything except get eaten. In technical terms, she's actually just a prop (sorry Grandma).

12. Click on the "New" button at the bottom of the *Characters* list. Name the character "Red's mom".
13. In the *Edit Character* window, you should see a list of four traits. These are the shared traits that we created earlier. Note that the "Name" trait is set to match the name that we just gave the character, whereas the remaining three traits are set to "False" (see Figure 10).



Name	Value
Name	Red's mom
Possible victim	□
Possible rescuer	□
Armed	□

Figure 10: Traits for Red's mom

14. Since Red's mom is also a possible victim, but is not a possible rescuer, and is not armed, set "Possible victim" to "True", and leave the other two traits as they are.
15. Now click "Ok" to go back to the main window.

We now have our two victims. Next we'll create the wolf.

16. Click on the "New" button at the bottom of the *Characters* list, and name the new character "Big Bad Wolf".
17. The wolf is neither a possible victim nor a possible rescuer, and is initially unarmed, so leave all the traits as they are, and click "Ok".

Finally, we'll create our two possible rescuers.

18. Create a new character named "Hunter".
19. The Hunter is a possible rescuer, so check the check-box in the "Value" column for "Possible rescuer", and then click "Ok".
20. Create a new character named "Policeman".
21. The Policeman is also a possible rescuer, so check the "Possible rescuer" checkbox, and click "Ok".

Now we have all our characters.

Note that there are 2 big "up" and "down" arrows to the right of the *Characters* list. This allows you to move characters up and down in the list. The order of characters isn't important to **WideRuled**, so you can use this to keep the characters sorted in an order that helps you to keep track of the story. For example, you may want to put all the possible victims together, so that you can remember who they are.

Relationships

In addition to Traits, characters can also have Relationships, which determine how characters are related to each other. Each of these relationships also has a “strength”, which determines how strong that relationship is. For our example we will not be using relationships.

Environments

We will also not be using Environments in our example. Environments are similar to characters, except that they represent the settings in which a story takes place, rather than the characters involved in the story. As with Characters, Environments can have both Traits and Relationships.

4. Creating Author Goals and Plot Fragments

Now that we have created our characters, we need to start to define what happens in the story. The events in a story are represented by *Author Goals*. The Author Goals represent the events that the author wants to happen within a story. These goals are defined in a specific order, and **all** must be satisfied for the story to succeed.

Each Author Goal can contain a number of *Plot Fragments*, which represent different ways in which the Author Goal can be satisfied. When an Author Goal is started, **WideRuled** will check the Plot Fragments for the Author Goal, and choose one. Each Plot Fragment contains a set of *Preconditions*, which **WideRuled** will check, in order, before the Plot Fragment is selected. If more than one Plot Fragment’s Preconditions are true, then one of the matching Plot Fragments will be chosen randomly.

In addition to preconditions, a Plot Fragment contains a series of *Story Actions*, which are executed, in order, when a Plot Fragment is chosen. There are six types of Story Actions available in **WideRuled**:

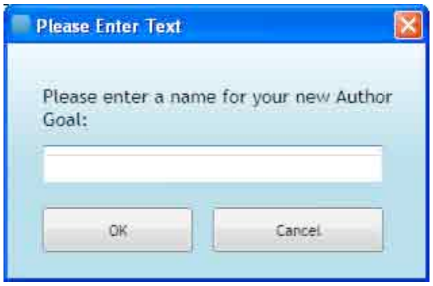
- Output Text: displays some text in the story
- Pursue a Subgoal: start pursuing an Author Goal
- Make a calculation
- Create a new Character/Environment/Plot Point
- Edit a saved Character/Environment/Plot Point
- Delete a saved Character/Environment/Plot Point

Once all the Story Actions in a Plot Fragment have been executed, the Author Goal which contains the Plot Fragment is considered satisfied.

Creating the Top-Level Author Goal

The first thing we need to do is create a top-level author goal, which will act as the start goal for the story. All **WideRuled** stories will have one starting goal, which is essentially to “tell the story”.

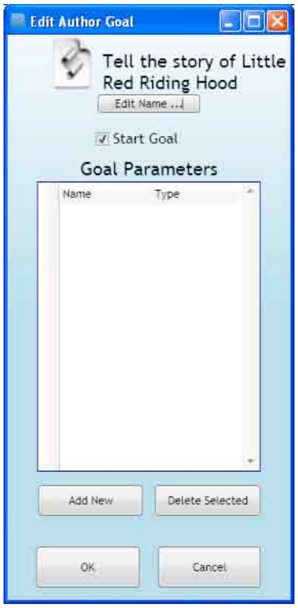
1. Click on the “New” button at the bottom of the *Goals and Plot Fragments* list of the main window. A window titled *Please Enter Text* will appear (see Figure 11). Enter the text “Tell the story of Little Red Riding Hood”, and press “Ok”.



Please enter a name for your new Author Goal:

Figure 11: Creating a new Author Goal.

2. You will now see the *Edit Author Goal* window (see Figure 12). This window lets you edit the name of the Author Goal, specify whether it's the start goal, and create a list of *Goal Parameters*. Goal Parameters are values that can be passed in from a containing Plot Fragment, which allows for re-use of common Author Goals. We won't be using parameters in our simple example.


Edit Author Goal

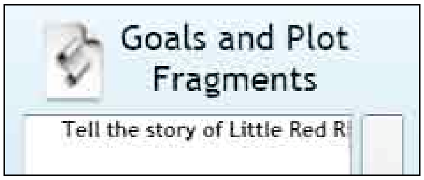
Tell the story of Little Red Riding Hood

☑ Start Goal

Goal Parameters

Figure 12: Edit Author Goal window.

3. Leave the "Start Goal" check-box checked, and click on the "Ok" button at the bottom of the window. The window will close, and you should see your first Author Goal in the *Goals and Plot Fragments* list of the main window (see Figure 13).

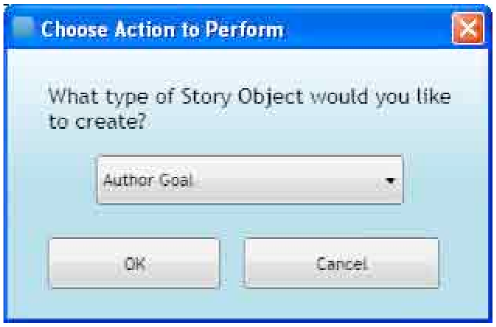

Goals and Plot Fragments

Tell the story of Little Red R

Figure 13: The first Author Goal.

For an Author Goal to be satisfied, there must be at least one Plot Fragment. At this point, we only want to specify one way of satisfying our top-level Author Goal, which is to tell the story! To do this, we will now create one Plot Fragment, with no preconditions, called "Tell the story". This Plot Fragment will contain the sequence of Author Goals for our story.

4. Click on the "New…" button at the bottom of the *Goals and Plot Fragments* list of the main window. This time, since we already have an Author Goal, we are given a choice of creating either another Author Goal, or a Plot Fragment. You should see a window titled "Choose Action to Perform", with a pull-down that can be used to select either "Author Goal" or "Plot Fragment" (see Figure 14). Choose "Plot Fragment", and click "Ok".

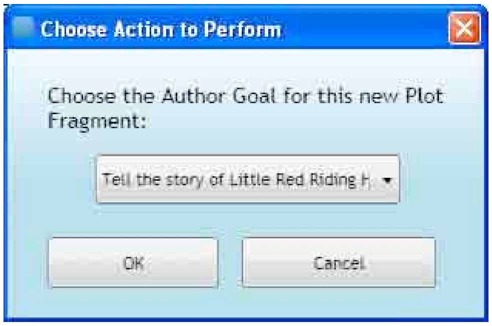


What type of Story Object would you like to create?

Author Goal

Figure 14: Creating a new Plot Fragment.

5. You should now see another window titled "Choose Action to Perform", asking you to choose the Author Goal for the new Plot Fragment (see Figure 15). We want the Plot Fragment to be attached to our initial goal, "Tell the story of Little Red Riding Hood", so choose this goal, and click "Ok'.

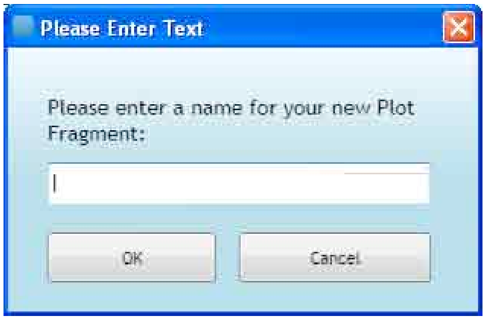


Choose the Author Goal for this new Plot Fragment:

Tell the story of Little Red Riding H

Figure 15: Attaching Plot Fragment to an Author Goal.

6. The final step before you can edit the new Plot Fragment is to give it a name (see Figure 16). Enter the name of the Plot Fragment, "Tell the story", and click "Ok".



Please enter a name for your new Plot Fragment:

Figure 16: Name the new Plot Fragment.

7. You should now see the *Edit Plot Fragment* window (Figure 17).

The *Edit Plot Fragment* window lets you add, edit and delete the Preconditions (see B in Figure 17) and the Story Actions (see C in Figure 17) for the Plot Fragment. You can also change the name of the Plot Fragment, specify a list of Parameters (see A in Figure 17), and change the Author Goal that is associated with the Plot Fragment (see D in Figure 17).

As our first Author Goal, "Tell the story of Little Red Riding Hood", only has one Plot Fragment, which must always be chosen, we will not create any Preconditions. Instead, we will create a series of Story Actions, which represent various Subgoals that constitute the events in the story.

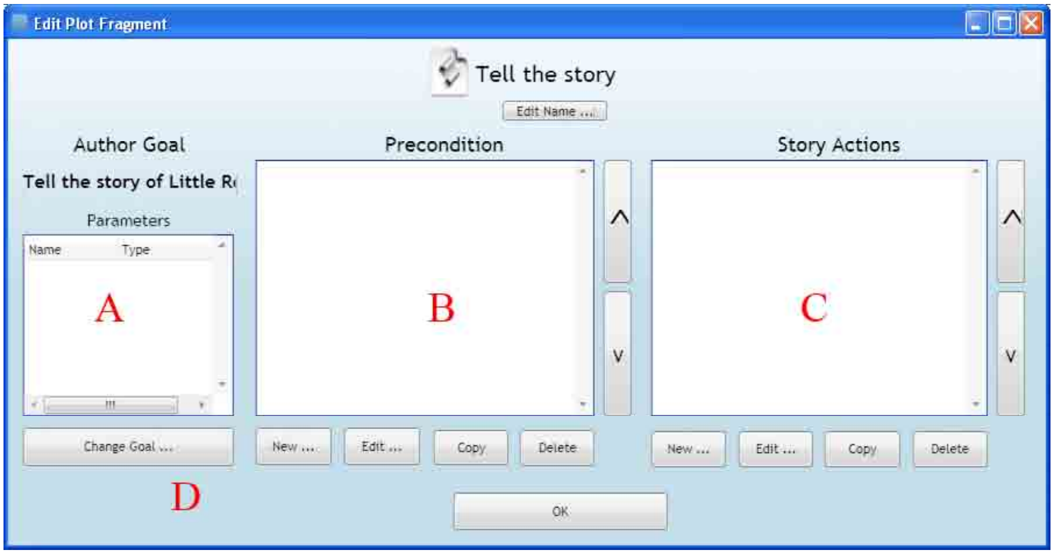
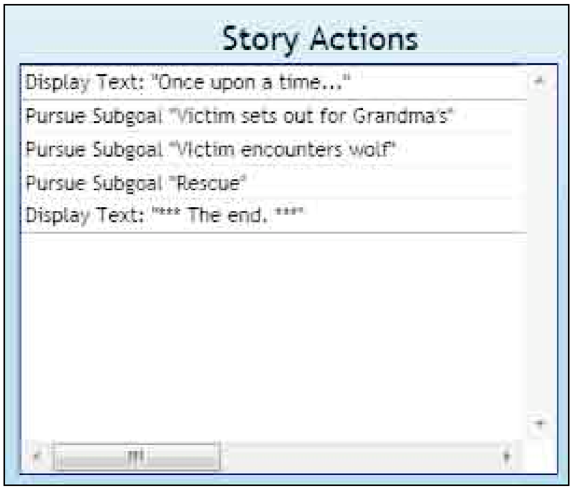


Figure 17: The *Edit Plot Fragment* window.

Our final story will consist of 3 Subgoals, as shown in Figure 18. We also want to print some text at the start and end of the story. We'll start be creating these text messages, and then create the subgoals. Finally, we will add the subgoals to our Plot Fragment.



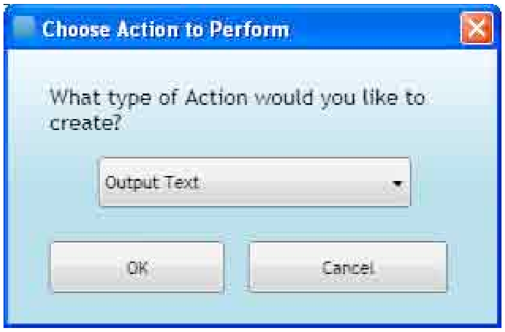
Display Text: "Once upon a time..."
Pursue Subgoal "Victim sets out for Grandma's"
Pursue Subgoal "Victim encounters wolf"
Pursue Subgoal "Rescue"
Display Text: "*** The end. ***"

Figure 18: Story actions for the "Tell the story" Plot Fragment.

Displaying text

Now we'll create the initial message, "Once upon a time…"

8. Click on the "New…" button at the bottom of the Story Actions list in the Edit Plot Fragment window. You should see a window titled *Choose Action to Perform*" (see Figure 19). This lets you select the action to be performed. Choose "Output Text", and press "Ok".

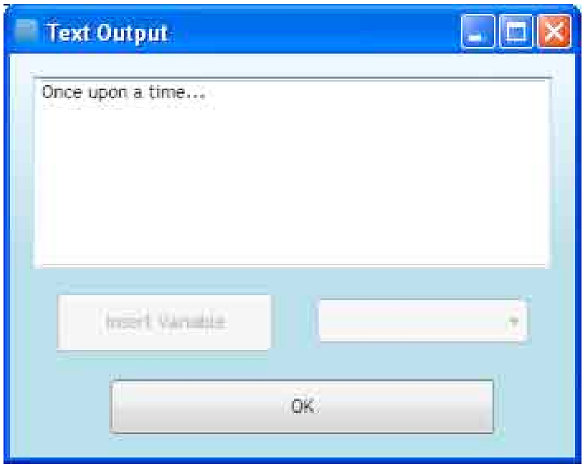


What type of Action would you like to create?

Output Text

Figure 19: Choosing an action for a Story Action.

9. You should now see a window titled “Text Output”, where you can edit the text to be displayed for the Story Action (see Figure 20). Enter the text “Once upon a time…”, and click “Ok”.

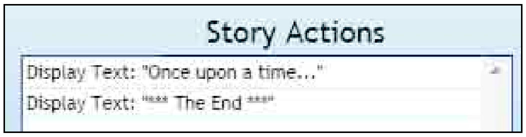


Once upon a time…

Figure 20: Editing the text for the Output Text story action.

10. Now create our second Output Text action, by creating another Story Action, choosing the “Output Text” action, and entering the text “*** The End ***”.

We now have two Story Actions in our Plot Fragment (see Figure 21).



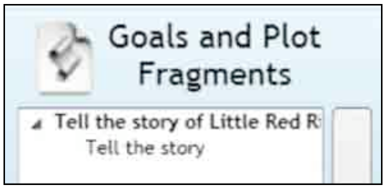
Display Text: "Once upon a time…"

Display Text: "*** The End ***"

Figure 21: Initial Story Actions for the “Tell the story” Plot Fragment.

11. Click the “Ok” button at the bottom of the Edit Plot Fragment window.

You should see one Author Goal in the *Goals and Plot Fragments* list, “Tell the story of Little Red Riding Hood”, with one Plot Fragment, below and indented slightly, “Tell the story” (see Figure 22). Notice that there is a small arrow at the left of the Author Goal. You can click this to expand or collapse the list of contained Plot Fragments. This will be very useful once your story becomes more complex.



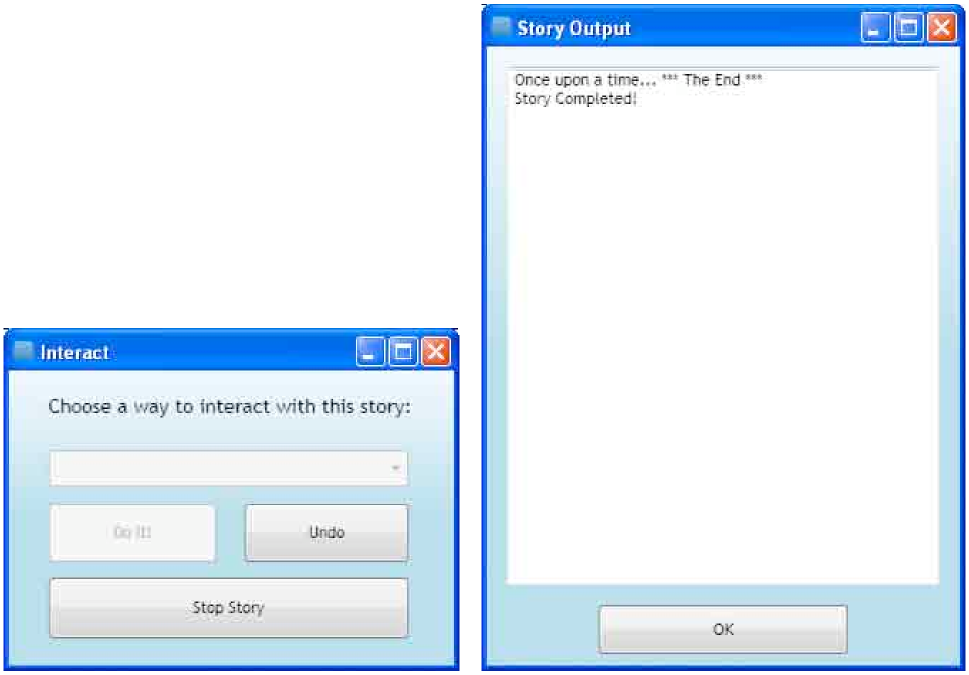
Tell the story of Little Red R
Tell the story

Figure 22: The initial Author Goal and contained Plot Fragment.

Testing the story

At this point, we actually have a complete (although rather trivial) story. Before creating the Subgoals that will form the actual story, we can test out our initial Author Goal.

12. Click on “Generate Story”. You should see two windows: “Interact” and “Story Output” (see Figure 23). For now, ignore the “Interact” window – this is used to trigger Author Goals based on user interaction. We’ll come back to this in Part II of the tutorial.



```
Once upon a time... *** The End ***
Story Completed!
```

Figure 23: Interact and Story Output window.

As you can see, the Story Actions in the one Plot Fragment are executed, printing out our two messages.

5. Creating the Subgoals

Now that we have our main Author Goal, we need to create each of the Subgoals for the story. The first Subgoal is “Victim sets out for Grandma’s”.

1. Click on the “New…” button at the bottom of the Goals and Plot Fragments list, and create a new Author Goal.
2. Name the new Author Goal “Victim sets out for Grandma’s”.

Creating the Plot Fragment

Now we need to create the Plot Fragment for this Author Goal. There will only be one way in which this Author Goal can be accomplished, by having a victim walk through the forest.

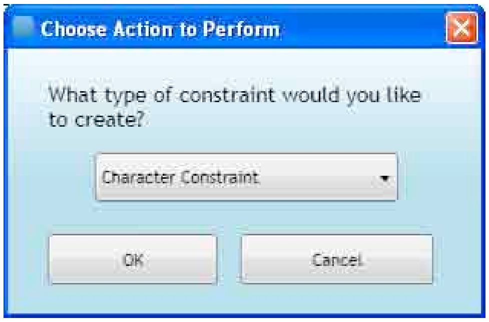
3. Click on the “New…” button at the bottom of the Goals and Plot Fragments list, and create a new Plot Fragment.
4. Choose “Victim sets out for Grandma’s” as the Author Goal for our new Plot Fragment.
5. Name the new Plot Fragment “Walking through the forest”.

Creating Preconditions

Now we need to set the Precondition for this Plot Fragment. An obvious Precondition for the Author Goal “Victim sets out for Grandma’s” is that there is a possible victim. We want to allow for either of our possible victims, Little Red Riding Hood or Red’s mom, to be the victim. To do this, we will carefully specify the *Precondition* for the Plot Fragment such that **WideRuled** can choose either of these two characters.

Preconditions can be used to constrain which Plot Fragments are chosen by **WideRuled** to satisfy an Author Goal. When you specify a set of Preconditions, **WideRuled** will do its best to match those preconditions. We can take advantage of this to allow for variation in the ways in which the story is told. Now lets make our Precondition.

6. Click on the “New…” button at the bottom of the Preconditions list in the Edit Plot Fragment window. You should see a window titled “Choose Action to Perform” (see Figure 24). You can create either a Character Constraint or an Environment Constraint.



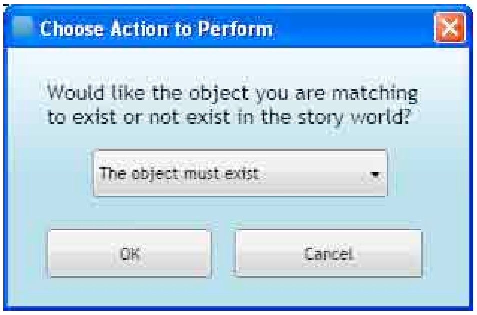
Choose Action to Perform

What type of constraint would you like to create?

Character Constraint

Figure 24: Creating a Character Constraint.

7. Choose to create a “Character Constraint”, and click on “Ok”. This will let you specific constraints based on the traits or relationships of characters.
8. You will now see a window titled “Choose Action to Perform”, which asks you whether the object you are matching should exist or not exist in the world (see Figure 25).



Choose Action to Perform

Would like the object you are matching to exist or not exist in the story world?

The object must exist

Figure 25: Choosing whether the object must exist or not exist.

Its possible to set a Precondition such that the Plot Fragment will only be chosen if there *does not exist* an object that matches your constraints. In our case, we want to set a constraint such that there *does exist* a character that is a potential victim.

9. Select “The object must exist”, and click on “Ok”. You will now see the Edit Precondition Statement window (See Figure 26).

The Edit Precondition window contains a list of Constraints, all of which must be satisfied for the Precondition to be True. Each Constraint has a set of Details, which can be edited when the Constraint is selected in the Constraints list.

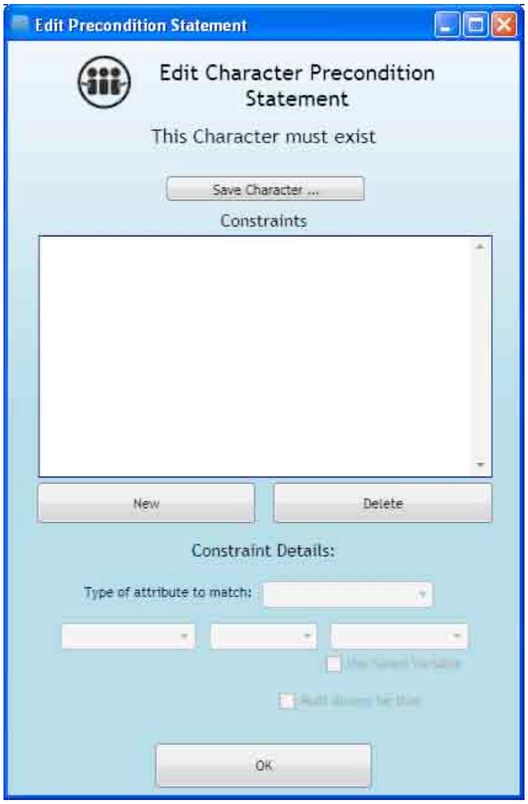
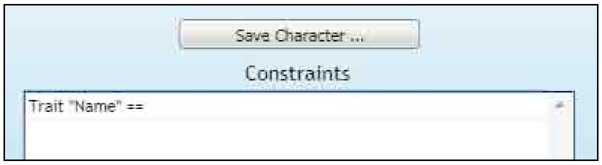

Edit Precondition Statement
Edit Character Precondition Statement
This Character must exist
Save Character ...
Constraints
New
Delete
Constraint Details:
Type of attribute to match:
OK

Figure 26: The Edit Precondition Statement window.

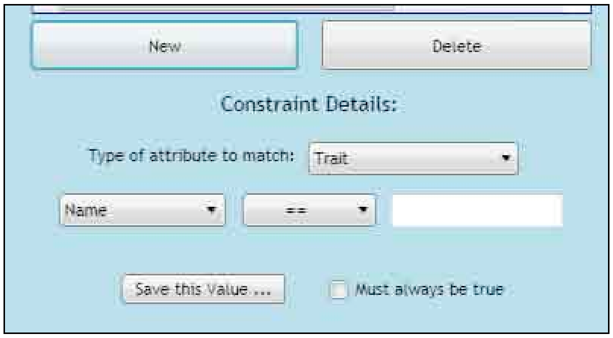
Now we want to set our precondition, that there must be a possible victim. To do this, we can use the “Possible victim” trait that we created for characters, and specify that there must be a character with this trait set to “True”.

10. Click on the “New” button at the bottom of the Constraints list. A new Constraint will be created (see Figure 27), and the Constraint Details section of the window will be enabled (see Figure 28).



Trait "Name" ==

Figure 27: A new constraint is added.



Type of attribute to match: Trait

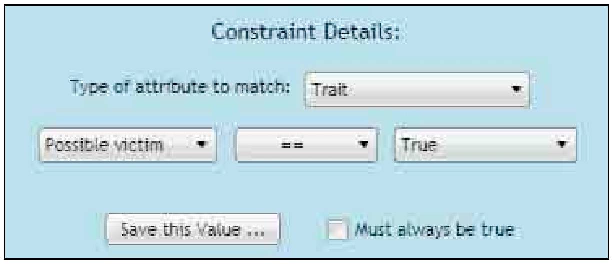
Name ==

Figure 28: Constraint details.

We now want to set the details for this Constraint. We want to specify that the Trait called “Possible victim” should be equal to “True”. The Constraint Details lets you specify which trait or relationship to match, and lets you construct a simple *expression* to determine whether or not the constraint is satisfied. This expression consists of the *name* of the trait or relationship, an *operator*, and the value to be matched.

The operator can either be == (equals), != (not equals), or a special operator, “save only”. If you choose “save only”, the trait won’t be matched. Instead, it will be kept around for use in the Story Actions for this Precondition. We’ll use this in a moment. But first, we need to specify our Constraint.

11. Make sure that the “Type of attribute to match” is set to “Trait”.
12. In the next line, make sure that the name of the attribute is set to “Possible victim”, that the operator pull-down is set to “==”, and that the final field in the Condition is set to “True” (see Figure 29).



Type of attribute to match: Trait

Possible victim == True

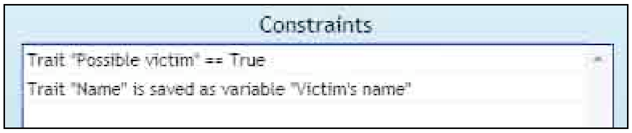
Figure 29: Constraint details for the first Precondition.

Now we have our constraint. However, we also need to be able to make use of the victim’s name, so that we can display the name to the reader. We also want to be able to remember this name later in the story, so that the victim remains constant. There will be a breakdown of consistency in our narrative if the victim keeps changing from one Author Goal to the next!

To remember the name of the victim, we will use the special “save only” operator.

13. Create a new constraint, and set the trait to “Name”.
14. Set the operator to be “Save only”. A window named “Please Enter Text” will appear, asking you to name the variable for this saved value (see Figure 30). This is the variable we will use to later retrieve the name of the victim.
15. Give the variable the name “Victim’s name”, and click “Ok”.

You should now have two constraints, as shown in Figure 30.

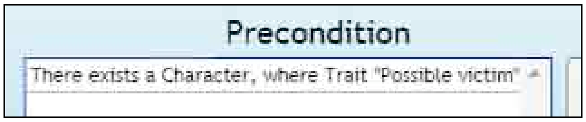


Trait "Possible victim" == True

Trait "Name" is saved as variable "Victim's name"

Figure 30: Constraints for the first Precondition.

16. Finish editing this Precondition by clicking on the “Ok” button at the bottom of the Edit Precondition window. In the Preconditions list of the Edit Plot Fragment window, you should have one precondition (see Figure 31).



There exists a Character, where Trait "Possible victim"

Figure 31: Precondition for “Walking through the forest”.

Creating Story Actions

Now that we have our Precondition, we need to create the Story Actions for the “Walking through the forest” Plot Fragment. The first action we will create will display some text to the reader, telling her what is happening in this Plot Fragment.

17. Click on the “New…” button at the bottom of the Story Actions list, and choose an “Output Text” as the new action.

Note that the “Insert Variable” button at the bottom of the Text Output window is enabled, and the pull-down menu contains the name of the variable we created in our Precondition. This allows us to embed the value of the variable into our text. We’ll use this to display the name of the victim.

18. Click on the “Insert Variable” button. The text “<Victim’s Name>” will appear in the Text Output window. This tells **WideRuled** to substitute this text with the value of the variable “Victim’s Name”, ie. the name of the victim chosen in our Constraint.
19. Click in the Text Output text editing area after the text “<Victim’s Name>”, and type the following text:

    was walking through the forest to deliver some food to Grandma.

20. Click “Ok”.

We now have our first Story Action. At this point, if we run the story, the reader will see the text above, with the variable substituted with the name of one of the two possible victims.

However, there’s a problem. Variables created in a Precondition only exist as long as a Plot Fragment is being executed. After the last Story Action is completed, the variable will disappear. This means that **WideRuled** will forget who the victim is. If we use a Precondition in a later Plot Fragment to choose a victim, it may or may not be the same victim.

To deal with this, we need to use a Plot Point. We’ll come back to our Plot Fragment in a moment.

21. Click “Ok” in the Edit Plot Fragment window, to go back to the main **WideRuled** window.

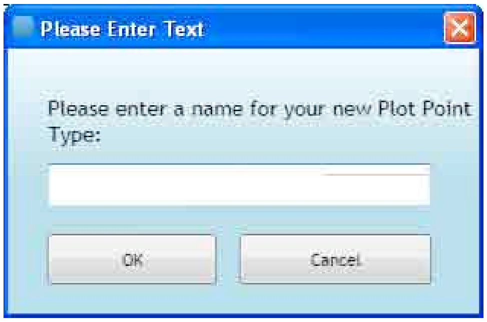
Creating Plot Points

A Plot Point is an important piece of information within a story, which can be saved and then later retrieved or changed. In our story, we initially need one Plot Point: Victim Chosen. We will use this Plot Point to store the name of the victim, so that later in the story we can retrieve the correct character to act as the victim.

In the main **WideRuled** window, there is a Plot Point Types list. This lists all the potential Plot Point Types in your story. Note that this is just a list of the types of Plot Points, not the actual Plot Points. We'll need to create an *instance* of a Plot Point in our Plot Fragment when our victim is chosen.

We'll start by creating a Plot Point Type for our victim.

22. Click the "New…" button at the bottom of the Plot Point Type list. You will see a window named "Please Enter Text", where you can name your Plot Point Type (see Figure 32). Name your Plot Point Type "Victim Chosen", and click "Ok".



Please enter a name for your new Plot Point Type:

Figure 32: Naming your Plot Point Type.

Now you will see a window named "Edit Plot Point Type", where we can edit our new Plot Point Type (see Figure 33). This window contains a list of traits that can be stored. Each has a name and a type (text, number or True/False.

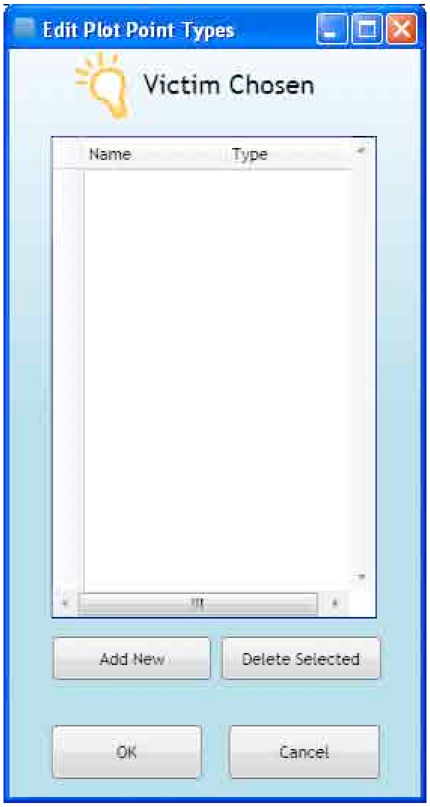
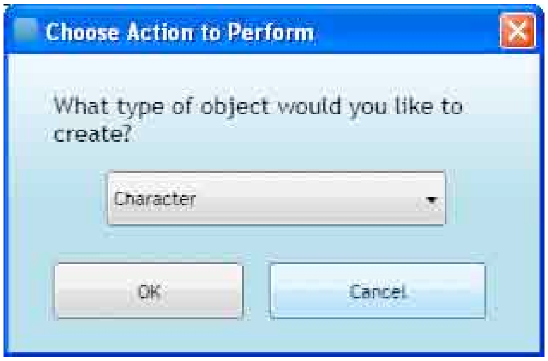


Figure 33: Edit Plot Point Type window.

23. Click the “Add New” button at the bottom of the list to add a new trait. The trait will appear in the list, with name “New Trait”, and type “Text”.
24. Set the name of the trait to “Name”, and the type to “Text”. We’ll use this trait to store the name of the victim.
25. Click on the “Ok” button to close the window.

We now have one Plot Point Type, which we can use in our Plot Fragment to remember who our victim is. We’ll now go back to the Plot Fragment, and add in Story Actions to store the victim’s name in our Plot Point.

26. Double-click on the “Walking through the forest” Plot Fragment in the Goals and Plot Fragments list, to open the Edit Plot Fragment window.
27. Click on the “New…” button at the bottom of the Story Actions list to create a new Story Action.
28. Choose “Create a new Character/Environment/Plot Point”, and click “Ok”.
29. You will see a window named “Choose Action to Perform”, asking you to choose the type of object you want to create (see Figure 34). Choose “Victim Chosen Plot Point”, and click “Ok”.

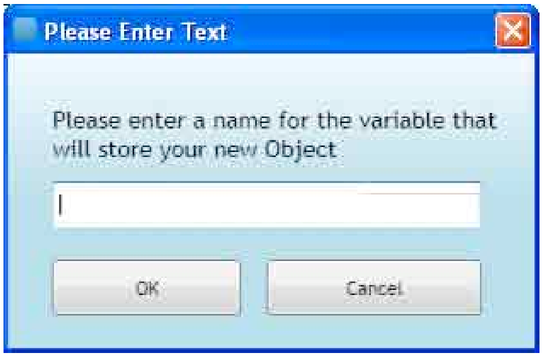


What type of object would you like to create?

Character

Figure 34: Creating a Plot Point

30. You will see a window named “Please Enter Text”, asking you to enter a name for the variable to store your new object (see Figure 35). Name the variable “The victim”.



Please enter a name for the variable that will store your new Object

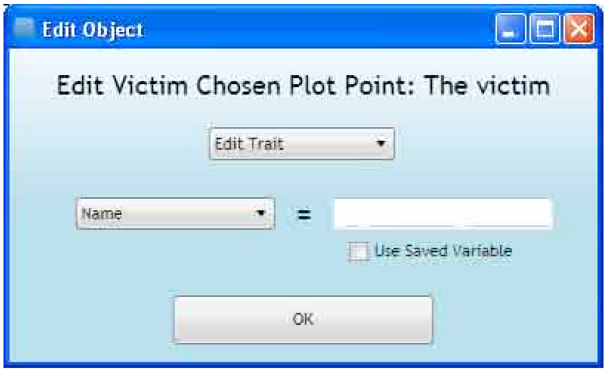
Figure 35: Naming your new Plot Point.

The new Plot Point will be shown in a window named Edit Plot Point. Note that the “Name” trait is listed, with no value. You can enter a value here. However, in our story we actually want to store the value saved in our “Victim’s name” variable, so that it can be accessed from other Plot Fragments. We’ll need to use a Story Action to do this.

31. Close the Edit Plot Point window by clicking on “Ok”.
32. Create a new Story Action of type “Edit a saved Character/Environment/Plot Point”, and chose to edit an object of type “Victim Chosen Plot Point”.

33. You will now see a window titled “Choose Action to Perform”, asking you which saved “Victim Chosen Plot Point” you would like to edit. As we’ve only saved one of this type of Plot Point, there is only one choice, “The victim”. Choose this saved plot point, and click “Ok”.

You will now see a window titled “Edit Object” see Figure 36). This window lets you change the traits of a saved plot point. The first pulldown menu lets you choose what you want to do to the saved Plot Point. The second pulldown menu lets you choose which trait you want to edit.



Edit Object

Edit Victim Chosen Plot Point: The victim

Edit Trait

Name =

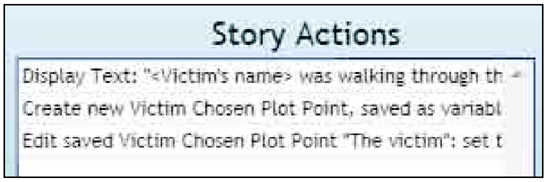
Figure 36: Editing a saved Plot Point.

34. Make sure that the first pulldown menu is showing “Edit Trait”, and the second is showing “Name”.

A trait can be set to a constant value, by typing in some value, or can be set to the value of a variable. We want to set the “Name” trait to the value of our “Victim’s Name” variable.

35. Check the “Use Saved Variable” checkbox. The entry field beside the “Name” pulldown should turn into a pulldown menu, listing all available variables. In our current Plot Fragment, there is only one variable, “Victim’s name”, so this is shown. Make sure it is selected, and click “Ok”.

Your Plot Fragment should now have three Story Actions, as shown in Figure 37.



Story Actions

Display Text: "<Victim's name> was walking through th

Create new Victim Chosen Plot Point, saved as variabl

Edit saved Victim Chosen Plot Point "The victim": set t

Figure 37: Story actions for Plot Fragment “Walking through the forest”.

Our Plot Fragment is now complete. We have set a Precondition that there must be a victim, we have displayed some text to the reader, and we have stored the name of the victim in a Plot Point.

36. In the Edit Plot Fragment window, click “Ok” to close the window.

Connecting an Author Goal to a Plot Fragment as a Subgoal

So far, we have our “Victim sets out for Grandma’s” Author Goal, and we have our Plot Fragment, “Walking through the forest”, which can be used to satisfy this goal. However, we haven’t specified when this Author Goal must be satisfied. What we need to do now is attach this Author Goal to our earlier Plot Fragment, “Tell the story”.

37. Double-click on the “Tell the story” Plot Fragment in the Goals and Plot Fragments list, to open the Edit Plot Fragment window.
38. Click on the “New…” button at the bottom of the Story Actions list to create a new Story Action.
39. Choose type “Pursue a subgoal”, and click “Ok”.
40. You will see a window named “Choose Action to Perform”, which asks you to choose an Author Goal to pursue (see Figure 38). Choose the Author Goal “Victim sets out for Grandma’s”., and click “Ok”.

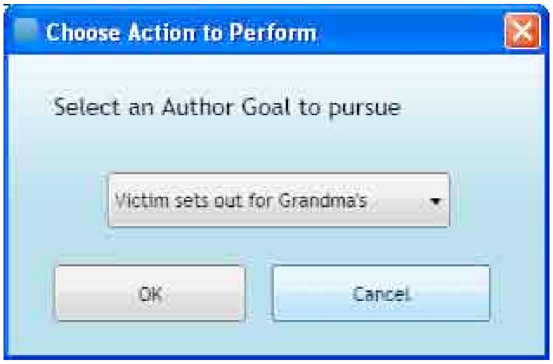
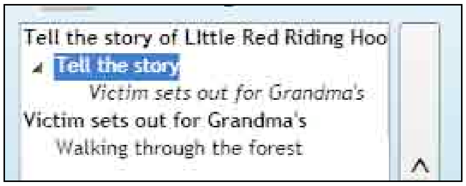

Choose Action to Perform
Select an Author Goal to pursue
Victim sets out for Grandma's
OK
Cancel

Figure 38: Selecting an Author Goal to pursue.

Notice that we now have a new Story Action in our list of actions. However, the “Pursue Subgoal” action is after the “Display text: *** The End ***” action, which isn’t what we want. Story actions are executed in order, from top to bottom, so we need to move our new action up above the end of the story.

41. Select “Pursue Subgoal: Victim sets out for Grandma’s” in the list, and click on the big “up” button to the right of the list. The selected Story Action should move up one place in the list.
42. Click “Ok” to close the Edit Plot Fragment window.

We now have a top level Author Goal, “Tell the story of Little Red Riding Hood”, which contains a single Plot Fragment, “Tell the story”. This Plot Fragment contains one Subgoal, “Victim sets out for Grandma’s” (see Figure 39).



Tell the story of LIttle Red Riding Hoo
Tell the story
Victim sets out for Grandma's
Victim sets out for Grandma's
Walking through the forest

Figure 39: Goals and Subgoals for our story so far.

We can now test out our story, and see how **WideRuled** attempts to satisfy the Preconditions for a Plot Fragment.

43. Click “Generate Story” to run the story.

If you run the story several times, you should see two different versions of the story, one in which Little Red Riding Hood is walking through the forest, and one in which Red’s mom is walking through the forest.

6. Adding Additional Subgoals

So far, we have created a set of characters, created Author Goals and Plot Fragments to express the author's intentions for the story, and created a Plot Point Type to store information about the story. Our simple story allows for variation in the choice of victim, based on satisfaction of Preconditions for a Plot Fragment.

In the rest of the tutorial, we will further explore the variations possible with Preconditions, in the form of choice of Rescuer, and will also introduce Interactivity, which will allow the reader to decide whether or not the Wolf has a gun.

We have two more Author Goals to add to the story, both of which will be Subgoals of our "Tell the story" Plot Fragment. The first goal, "Victim encounters wolf", will have one Plot Fragment. The second goal, "Rescue", will have several plot fragments, depending on which Possible Rescuer shows up on the scene.

Retrieving a Plot Point

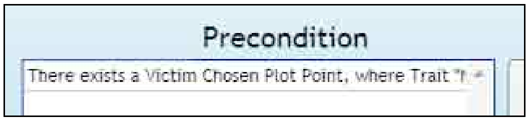
We'll start by adding the Author Goal "Victim encounters wolf". We'll need to retrieve the name of the victim from our Plot Point, and then use this name to tell the reader about the encounter. As there's only one Wolf, we won't need to set a Precondition for the Wolf character, although this is certainly possible.

1. First we need to create the Author Goal. Click on the "New…" button at the bottom of the Goals and Plot Fragments list in the main window, and create an Author Goal named "Victim encounters wolf". Click "Ok" in the Edit Author Goal window to go back to the main window.
2. Next, create a Plot Fragment named "Talking to wolf", with Author Goal "Victim encounters wolf". Don't close the Edit Plot Fragment window.
3. Create a new Precondition by clicking on the "New…" button at the bottom of the Precondition list in the Edit Plot Fragment window. This time, in addition to Character and Environment constraints, there is also an option to create a "Victim Chosen Plot Point Constraint". This is what we want, so choose this option. Specify that the object must exist.
4. In the Edit Precondition window, create a new Constraint.

We don't want to specify any specific constraints. As long as there exists a Victim Chosen Plot Point, that's all we are concerned about. What we want to do is retrieve the name of the victim. To do this, we can use the special "Save Only" operator in the Constraint Details.

5. In Constraint Details, make sure that the type of attribute is set to "Trait", and that the name of the trait is "Name". Choose the "Save Only" operator, and name the variable "Victim's name". This is the variable that we'll use to display the victim's name to the reader. Click "Ok" to close the Edit Precondition window.

You should now have one Precondition for your "Talking to wolf" Plot Fragment (see Figure 40). Now we need to display the name of the victim to the reader, along with information about this Plot Fragment.



Precondition

There exists a Victim Chosen Plot Point, where Trait "I

Figure 40: Retrieving the Victim Chosen Plot Point.

6. Create a new Story Action of type "Output Text", and enter the following text:

> In a clearing in the forest, **<Victim's name>** came across a wolf. The wolf asked where she was going, and she told him. Rushing ahead to Grandma's house, the wolf quickly disguised himself as Grandma, hoping to trick **<Victim's name>** into thinking he was Grandma. Upon reaching Grandma's house, **<Victim's name>** was shocked to see that Grandma was actually a wolf. Immediately, the wolf gobbled up **<Victim's name>**.

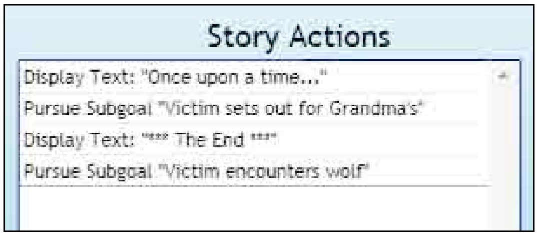
Note that where you see **<Victim's name>**, you should click on the "Insert variable" button to insert the variable that represents the victim's name.

7. After entering the text, press "Ok" to close the Text Output window. You should now have one Story Action, "Display Text", in your Plot Fragment.
8. Click "Ok" to close the Edit Plot Fragment window.

This Plot Fragment shows the reader what happens when the victim talks to the wolf. Note that there are actually several events within this Plot Fragment: talking to the wolf, the wolf going to Grandma's house, and the victim arriving at Grandma's house. We could have broken this down into several Author Goals, which would allow us to consider making alternate Plot Fragment or creating Preconditions to select alternative characters or environments for these goals. However, for simplicity and to save time in the tutorial, they are all grouped into one Plot Fragment.

Now that we have our Author Goal and Plot Fragment, we need to attach this Author Goal as a Subgoal to our top-level Author Goal's Plot Fragment, "Tell the story."

9. In the main window, double-click on the "Tell the story" Plot Fragment in the Goals and Plot Fragments list to edit the Plot Fragment.
10. Create a new Story Action, and choose type "Pursue a subgoal".
11. Select the "Victim encounters wolf" Author Goal, and click "Ok". The Author Goal should be added to the end of the list of Story Actions (see Figure 41).



Story Actions

Display Text: "Once upon a time..."

Pursue Subgoal "Victim sets out for Grandma's"

Display Text: "*** The End ***"

Pursue Subgoal "Victim encounters wolf"

Figure 41: New subgoal added to the "Tell the story" Plot Fragment.

Note that the new subgoal is at the end of the list, after the end of the story, which is obviously not what we want.

12. Select the new subgoal, and press the big "up" button to the right of the Story Action list to move the new subgoal above the "Display Text:" story action.
13. Click "Ok" to close the Edit Plot Fragment window.

You can now try running the story if you want. You should see the first part of the Little Red Riding Hood story, up to the point where the wolf eats the victim. We can't leave it like that – we now need to add in the rescuer.

Creating Alternative Plot Fragments

We'll now create the Author Goal for the arrival of the Rescuer. However, to make things more interesting, we'll allow for two possible Rescuers, the Hunter or the Policeman. We'll also provide different Plot Fragments depending on which rescuer arrives. The Hunter will kill the wolf, whereas the Policeman, being a Policeman, would rather arrest him.

We will need to remember who the Rescuer is, so lets first create a new Plot Point Type to store the name of the Rescuer.

14. Create a new Plot Point Type, by clicking on the “New…” button at the bottom of the Plot Point Types list in the main window.
15. Name the new Plot Point type “Rescuer chosen”.

As with the “Victim chosen” Plot Point Type, we just want to remember the name of the rescuer.

16. Add a new trait to the Plot Point Type by clicking on “Add New” at the bottom of the Edit Plot Point Type window. Name the new trait “Name”, and leave the Type as “Text”.
17. Click on “Ok” to close the Edit Plot Point Type window.

Next, we will create our new Author Goal for the rescue.

18. Create a new Author Goal, and name it “Rescue”.
19. Create a new Plot Fragment, with Author Goal “Rescue”, and name the Plot Fragment “Along comes the rescuer!”.

Now we need to let **WideRuled** choose the rescuer from the set of possible rescuers. We can do this with a Precondition, the same way that we chose the victim.

20. In the Edit Plot Fragment window, create a new Precondition. Make it a “Character Constraint”, and specify that the object must exist.
21. In the Edit Precondition window, create a new Constraint. Set the Trait to “Possible rescuer”, the operator to “==”, and the value to “True”. This will force **WideRuled** to choose a character that has the “Possible rescuer” trait set to “True”, ie. the Hunter or the Policeman.
22. We also need to remember the name of the Rescuer. Create another constraint, this time setting the Trait to “Name”, and the operator to “Save Only”. Name the variable “Rescuer’s name”.
23. Now click “Ok” to close the Edit Precondition window.

We now have our rescuer. As with the victim, we need to store the Rescuer in a Plot Point, which we created just now.

24. Create a new Story Action of type “Create a new Character/Environment/Plot Point Type”, and choose the type “Rescuer chosen Plot Point”. Name the Plot Point “The Rescuer”, and click “Ok” in the New Plot Point window.
25. Now we’ll store the name of the rescuer in our new Plot Point. Create a new Story Action of type “Edit a Saved Character/Environment/Plot Point”, and choose the type “Rescuer chosen Plot Point”. Select our new Plot Point. You should now be at the “Edit Object” window.
26. We want to save our “Rescuer’s name” variable into the “Name” trait of the Plot Point. Make sure “Edit Trait” is selected in the first pull-down menu. In the second line, make sure that “Name” is selected in the first pull-down menu, and then check the “Use Saved Variable” checkbox. There should only be one choice, “Rescuer’s name”. Make sure this is selected, and click the “Ok” button to close the window.

You should now have two Story actions, one to create the Plot Point, and one to edit the Plot Point and store the name of the Rescuer. Now we need to tell the reader who the rescuer is.

27. Create a new Story Action of type “Output Text”, and enter the following text, inserting the variable “Rescuer’s name” where you see **<Rescuer’s name>**:

    At that moment, the **<Rescuer's name>** happened to be walking past.

Now we have our Rescuer. But we haven't had the rescuer actually take action yet. Remember, we want the two rescuers to act differently. To do this, we will create a new Subgoal for this Plot Fragment, and we'll create two alternative Plot Fragments for that Author Goal.

28. Click "Ok" in the Edit Plot Fragment window to get back to the main window.
29. Create a new Author Goal, named "Rescuer takes action".
30. Create a new Plot Fragment, with Author Goal "Rescuer takes action", and name it "Hunter to the rescue".

Now we need to set a Precondition that states that the Hunter is the Rescuer, otherwise this Plot Fragment should not be chosen. To do this, we can set a Precondition based on the Rescuer chosen Plot Point that we created earlier.

31. Create a new Precondition, and make it a "Rescuer chosen Plot Point" constraint for which the object must exist.
32. Create a new Constraint, and make sure that the Trait selected is "Name", and the operator is "==".
33. We want the name to be "Hunter", so type in the text "Hunter" in the text field to the right of the "==" operator.
34. Click "Ok" to close the Edit Precondition window.

We now have our Hunter exists precondition. We also want to retrieve the name of the victim. We can do this in the same way that we did for our earlier Plot Fragments, by using the "Victim chosen" Plot Point.

35. Create a new Precondition, and make it a "Victim chosen Plot Point" constraint for which the object must exist.
36. Create a new Constraint, and make sure that the Trait selected is "Name".
37. Set the operator to "Save Only", and name the variable "Victim's name".

Now we have everything we need, so we can display the text to the reader.

38. Create a new Story Action of type "Output Text", and enter the following text, substituting variables as necessary:

    The hunter pulled out his axe and cut open the wolf. Grandma and **<Victim's name>** popped out, alive and well! They filled the wolf's stomache with stones, and boiled him, making a tasty wolf soup.

This completes our Plot Fragment.

39. Click "Ok" to go back to the main window.

Now we can do the same for the Policeman. The steps are almost the same as above.

40. Create a new Plot Fragment, with Author Goal "Rescuer takes action", and name it "Policeman to the rescue".
41. Create a new Precondition, and make it a "Rescuer chosen Plot Point" constraint for which the object must exist.
42. Create a new Constraint, and make sure that the Trait selected is "Name", and the operator is "==".
43. We want the name to be "Policeman", so type in the text "Policeman" in the text field to the right of the "==" operator.
44. Click "Ok" to close the Edit Precondition window.
45. Create a new Precondition, and make it a "Victim chosen Plot Point" constraint for which the object must exist.
46. Create a new Constraint, and make sure that the Trait selected is "Name".
47. Set the operator to "Save Only", and name the variable "Victim's name".

48. Create a new Story Action of type “Output Text”, and enter the following text, substituting variables as necessary:

    > The policeman, using his expert police training, induced vomiting in the wolf, and Grandma and **<Victim's name>** popped out. Quickly the Policeman arrested the wolf.

49. Click “Ok” to go back to the main window.

Now we need to add our “Rescuer takes action” Author Goal to our “Along comes the rescuer!” Plot Fragment, so that this Author Goal will be pursued.

50. Double-click on the “Along comes the rescuer!” Plot Fragment to edit it.
51. Create a new Story Action of type “Pursue Subgoal”, and chose the Author Goal “Rescuer takes action”.
52. Make sure that our new subgoal is at the end of the list of Story Actions.
53. Click “Ok” to close the Edit Plot Fragment window.

Finally, we need to go back to our top-level Plot Fragment, “Tell the story”, and add in the “Rescue” subgoal.

54. Double-click on the “Tell the story” Plot Fragment to edit it.
55. Create a new Story Action of type “Pursue Subgoal”, and chose the Author Goal “Rescue”.
56. Make sure that our new subgoal is after “Pursue Subgoal: ‘Victim encounters wolf’”, but before “Display text: ‘*** The End ***’”.
57. Click “Ok” to close the Edit Plot Fragment window.

And that’s it! Notice that in the Goals and Plot Fragments list in the main window, there are two Plot Fragments for the “Rescuer takes action” Author Goal. When the Author Goal is pursued, **WideRuled** will check each Plot Fragment to see if its Preconditions are satisfied. If only one Plot Fragment’s Preconditions are satisfied, then this Plot Fragment will be chosen. If, however, several Plot Fragments have their Preconditions satisfied, then one will be chosen randomly.

Try running your story. There should now be four possible versions of the story, depending on which of the two victims and which of the two rescuers are chosen.

7. Adding Interactivity

So far, we have created a procedurally generated story that allows for variation based on a set of *morphologies*. We haven’t, however, allowed the reader to make any choices. In the final part of the tutorial, we will add in a choice, in the form of an Author Goal that can be triggered at any time by the reader.

To make the story more interesting, what we will do is allow the reader to give the Wolf a gun, so that he has a chance to avoid being killed/arrested by the rescuer. We’ll do this by editing the Big Bad Wolf character, and setting his “Armed” trait to “True”.

Adding the Author Goal

First, we’ll create a new Author Goal and Plot Fragment to represent this event.

1. Create a new Author Goal, named “Give the wolf a gun”.
2. Next, create a new Plot Fragment, with Author Goal “Give the wolf a gun”, named “Arm the wolf”.

This Plot Fragment doesn’t really have any Preconditions. However, we need to retrieve the “Big Bad Wolf” character, so that we can edit its “Armed” trait. We’ll use a Precondition to do this.

3. Create a new Precondition, and make it a “Character” constraint for which the object must exist.
4. Create a new Constraint, and set its type of attribute to “Trait”, and the Name of the Trait to “Name”. Set the operator to “==”, and enter “Big Bad Wolf” in the text field.

This lets us retrieve the Big Bad Wolf. This time, we don’t want to save the Name in a variable, since we already know the name. Instead, we want to store the Character itself, so that in our Story Actions we can edit it. To do this, we, we can use the “Save Character” button at the top of the window.

5. Click on the “Save Character” button at the top of the Edit Precondition window.
6. You’ll see a “Please enter text” window that asks you for the name of the variable for this saved character. Name the variable “The Wolf”, and click “Ok”.
7. Now click “Ok” in the Edit Precondition window to close the window.

We now have our Wolf character stored in the variable “The Wolf”, so we can go ahead and edit its “Armed” trait, and tell the reader what happened.

8. Create a new Story Action of type “Output Text”, and enter the following text:

   The wolf has a valid gun license, and always carries at least a small pistol, if not a semi-automatic.

9. Create a new Story Action of type “Edit a saved Character/Environment/Plot Point”, and choose type “Character”. Choose to edit the saved character “The Wolf.” You will see a window named “Edit Object”.
10. Make sure that the first pull-down menu is set to “Edit Trait”.

This time, unlike with our Plot Points, we have several traits to choose from. We want to edit the “Armed” trait, and change its value to “True”.

11. In the second line, set the first pull-down menu, to the left of the “=”, to show “Armed”.
12. Set the second pull-down menu, to the right of the “=”, to show “True”.
13. Click “Ok”.
14. Finally, in the Edit Plot Fragment window, click “Ok” to go back to the main window.

Creating an Interaction Choice

We now have an Author Goal and an associated Plot Fragment which sets the Wolf to be armed. However, this Author Goal is not set as a Story Action in any Plot Fragments. Instead, we want to let the reader tell **WideRuled** to pursue this Author Goal. We can do this by using the “Interactivity Actions” button at the bottom of the main window.

15. Click on the “Interactivity Actions” button at the bottom of the main window. You should see a window named “Edit Interaction Choices” (see Figure 42).
16. Click on the “New…” button, and name the new Interaction Choice “Arm the Wolf”. Choose the Author Goal “Give the wolf a gun”. In the Pursue Subgoal window, click on the “Ok” button.
17. You should see a new Interaction Choice, “Arm the wolf”, listed in the “Edit Interaction Choices” window. Click “Ok” to go back to the main window.

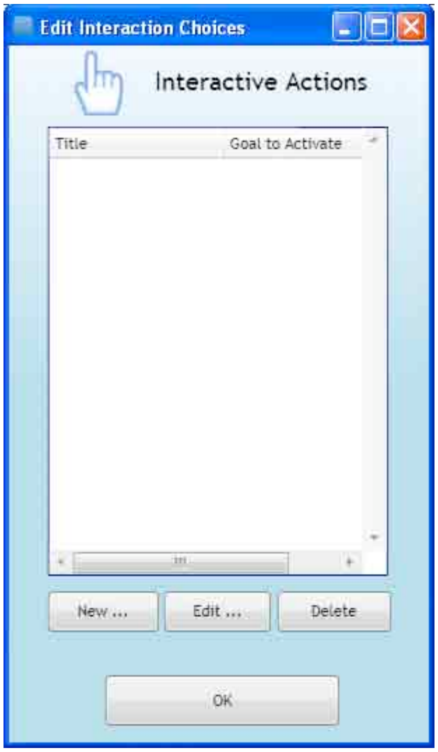


Figure 42: Edit Interaction Choices.

We now have our choice. If you run the story now, you should see that the “Arm the wolf” action is listed in the Interact window, and the “Do it!” button is now enabled. Try clicking the button. You will see the text “Interaction!”, followed by the text we entered in our “Arm the wolf” Plot Fragment, appear in the Story Output window, in the middle of our story.

Note, however, that this doesn’t seem to affect the story. This is because **WideRuled** doesn’t have any information about how the setting of the Wolf’s “Armed” trait to “True” should change the story. To do this, we need to go back into our “Rescuer takes action” Author Goal, and add a new Plot Fragment, which is only chosen when the Wolf is Armed.

Acting on the Reader’s Choice

First, we’ll create a new Plot Fragment for the “Rescuer takes action” Author goal, which has as the Wolf is Armed as a Precondition.

18. In the main window, create a new Plot Fragment, with Author Goal “Rescuer takes action”, and name it “Look out, he’s got a gun!”

As with our other two Rescuer Plot Fragments, we need to know who the victim is. However, unlike the other Plot Fragments, we don’t care who the Rescuer is, we just need to know their name. We’ll do this by creating two Preconditions, both of which simply match the appropriate Plot Point, and save the Name trait in a variable.

19. Create a new Precondition, and make it a “Victim chosen Plot Point” constraint for which the object must exist.
20. Create a new Constraint, and make sure that the Trait selected is “Name”.
21. Set the operator to “Save Only”, and name the variable “Victim’s name”.
22. Create a new Precondition, and make it a “Rescuer chosen Plot Point” constraint for which the object must exist.
23. Create a new Constraint, and make sure that the Trait selected is “Name”.
24. Set the operator to “Save Only”, and name the variable “Rescuer’s name”.

At this point, this Plot Fragment's Preconditions will always be satisfied. We need to add an additional, more restrictive Precondition, to check if the Wolf is Armed.

25. Create a new Precondition, and make it a "Character" constraint for which the object must exist.
26. Create a new Constraint, and make sure that the Trait selected is "Name".
27. Set the operator to "==", and enter the text "Big Bad Wolf" in the text entry field.

This Constraint tells **WideRuled** to match a character with name "Big Bad Wolf".

28. Create a new Constraint, and make sure that the Trait selected is "Armed".
29. Set the operator to "==", and set the pull-down menu to the right of the operator to "True".
30. Click "Ok" to go back to the Edit Plot Fragment window.

This Constraint asks **WideRuled** to match a character that, in addition to being named "Big Bad Wolf", also has the "Armed" trait set to "True". This will only be satisfied if the reader has made the choice to arm the wolf.

Now the only remaining step for this Plot Fragment is to tell the reader what happened.

31. Create a new Story Action of type "Output Text", and enter the following text, substituting variables as necessary:

    Unfortunately for the **<Rescuer's name>**, the wolf had a gun, which he pulled out just in time, and shot the **<Rescuer's name>** in the head. Grandma and **<Victim's name>** never managed to get out of the wolf's stomach.

We're almost done. However, at this point there's a problem. Although our new Plot Fragment will only be chosen when the Wolf's Armed trait is True, one of the other two Plot Fragments for this Author Goal, "Hunter to the rescue" and "Policeman to the rescue", will still be satisfied, since they don't have any Preconditions related to the Armed trait. Remember that if more than one Plot Fragment's Preconditions are satisfied, **WideRuled** will randomly choose one. That means in our current story, if the reader chooses to arm the wolf, there's a random chance that the wolf may or may not get to use his gun. This may seem realistic, but it reduces the feeling of agency on the part of the reader, so lets fix this.

What we need to do is go back to the two other Plot Fragments, and add in a Precondition that constrains the Plot Fragment only to be chosen if the wolf is **not** armed.

32. From the main window, double-click on the "Hunter to the rescue" Plot Fragment to edit it.
33. Create a new Precondition, and make it a "Character" constraint for which the object must exist.
34. Create a new Constraint, and make sure that the Trait selected is "Name".
35. Set the operator to "==", and enter the text "Big Bad Wolf" in the text entry field.
36. Create a new Constraint, and make sure that the Trait selected is "Armed".
37. Set the operator to "==", and set the pull-down menu to the right of the operator to "False".
38. Click "Ok" to go back to the Edit Plot Fragment window.
39. Click "Ok" to go back to the main window.

Note that this is almost exactly the same Precondition we put into our "Look out, he's got a gun!" Plot Fragment, except that now we are requiring the "Armed" trait to be "False" instead of "True".

Now we'll do the same for the "Policeman to the rescue" Plot Fragment.

40. From the main window, double-click on the "Policeman to the rescue" Plot Fragment to edit it.

41. Create a new Precondition, and make it a “Character” constraint for which the object must exist.
42. Create a new Constraint, and make sure that the Trait selected is “Name”.
43. Set the operator to “==”, and enter the text “Big Bad Wolf” in the text entry field.
44. Create a new Constraint, and make sure that the Trait selected is “Armed”.
45. Set the operator to “==”, and set the pull-down menu to the right of the operator to “False”.
46. Click “Ok” to go back to the Edit Plot Fragment window.
47. Click “Ok” to go back to the main window.

And that’s it! Run the story. You should be able to choose the “Arm the wolf” choice, and see the story change accordingly.

8. Finishing up

This is a very basic story. We’ve created 5 characters, a series of Author Goals, and variations in the Plot Fragments based on which victim and rescuer were selected by **WideRuled**. We’ve also added an Interaction Choice, which lets the reader give the wolf a gun.

There are several features of **WideRuled** that we haven’t looked at, including Environments, Relationships, and Parameters. Environments are very similar to Characters, except they are intended to represent locations. Relationships can be given to both Characters and Environments. They are similar to Traits, except that they list another Character or Environment, and have a “strength” value that represents how strong the relationship is. An example of a relationship would be “loves”, meaning that one character loves another. You can test the strength of a relationship in a Precondition, and edit the value of a Relationship in a Story Action.

To find out more about using **WideRuled**, have a look at the example story that comes with the software. There is additional information about how to use the tool at James Skorupski’s wiki: http://www.soe.ucsc.edu/~jskorups/wiki/doku.php?id=wide_ruled:wide_ruled_v2.