

The HypeDyn Hypertext Fiction Editor

Tutorial 1: Nodes, Links and Rules

Contents

1	Introduction	1
2	Getting started	3
3	Creating a text node	3
4	Adding text to a node	4
5	Creating links between nodes	5
5.1	Creating the other nodes	5
5.2	Creating a link	6
5.3	Testing the link	9
5.4	Creating more links	10
6	Creating rules with conditions	11
6.1	Creating the node	12
6.2	Adding the condition	13
6.3	Testing the rule	14
7	Reading the story	15
8	Conclusion	15

1 Introduction

In this tutorial, we will build a basic interactive story in the HypeDyn hypertext fiction editor. HypeDyn is a tool which lets you create “hypertext”, a form of writing which usually consists of:

- A collection of text fragments or “nodes”, containing some text, which are normally displayed one at a time in a hypertext reader;

- “links” between these nodes, which provide a connection from one node, or more commonly from a section of text in one node, to another node. A reader is usually able to “follow” a link to its destination node by, for example, clicking on the link;
- “rules” which can be attached to a link, which contain a set of conditions, and a set of actions, such as following a link, which are performed if the rule’s conditions are satisfied.

In this tutorial, we will be creating a simple hypertext *fiction*. Hypertext fiction usually consists of a story told in hypertext. Most well-known examples of hypertext fiction let the reader explore through a set of text fragments, encouraging them to discover different paths through the story. These alternate paths sometimes uncover new information which the reader didn’t previously know, thereby altering their perception of the story.

Our example, “Little Red Riding Hood”, involves the reader exploring several paths through a story. The main path tells a simplified version of the traditional folk tale, about a little girl with a red hood who meets a wolf on the way to visit her Grandmother. An additional side path, which can only be accessed once the reader has visited the last node in the main path, reveals some additional background information about Red’s hood.

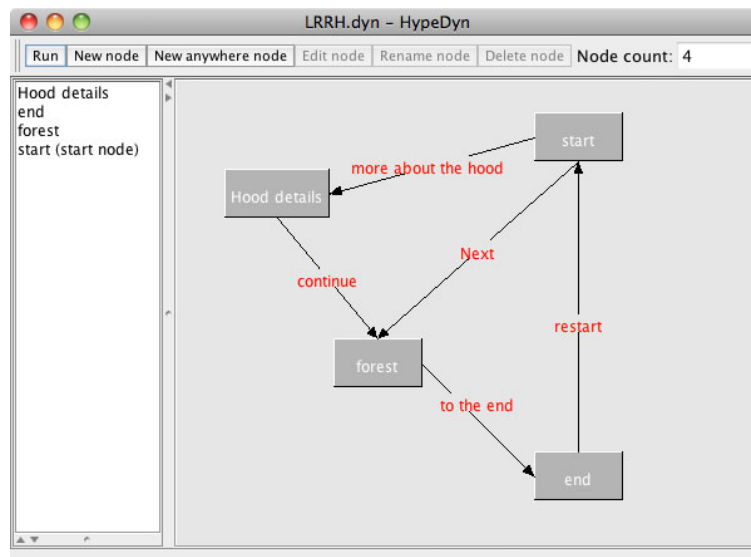


Figure 1: The completed “Little Red Riding Hood” story.

The nodes and links in the final story are shown in Figure 1. Although this is a very simple story, it will involve most of the basic functionality of HypeDyn.

Note: HypeDyn is a work-in-progress. If you encounter any errors, please report them as bugs on our Launchpad site: <https://launchpad.net/hypedyn>.

2 Getting started

First, open HypeDyn by double-clicking on the file **HypeDyn.exe** (in Windows) or **HypeDyn.app** (in MacOS).

Save your file by choosing Save from the File menu. Make sure that you give the file a **.dyn** extension.

3 Creating a text node

The first thing we need to do is to start creating some text nodes. We'll begin by creating the starting node, which represents the first fragment in the story. In this node, we will introduce the main character, Red, and the initial action of the story, the fact that she is walking through the forest.

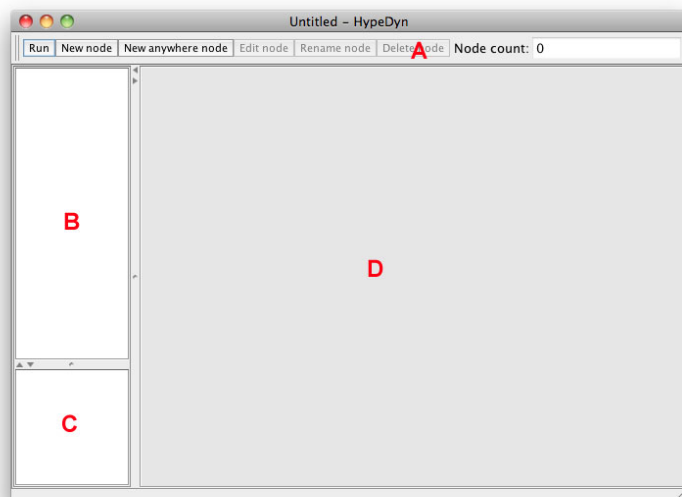


Figure 2: *The main HypeDyn interface.*

The main HypeDyn interface consists of the Toolbar (A), the Node list (B), the Fact list (C), and the Map view (D), as shown in Figure 2. The Toolbar contains the functions that we will use to create and edit our text nodes. The Node list contains a list of the nodes in your story. The Fact list contains a list of the facts in your story. The Map view displays the nodes and links. The Map view contains two different types of nodes: regular nodes and anywhere nodes. We will not be using facts or anywhere nodes in this tutorial.

1. To create a new text node, click on the New node button. A new text node will be created, and you will be asked to give it a title. Name the new text node “Start”, and click Ok, as shown in Figure 3.

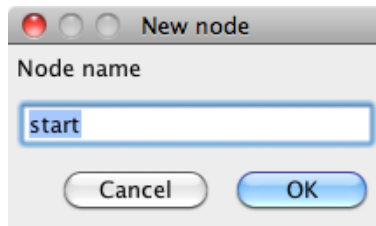


Figure 3: *Creating a new text node.*

Text nodes are the basic building blocks of hypertexts created with Hype-Dyn. Each text node has:

1. A title to identify the node
2. Some text content.

Once you've created a text node, you should see the node represented in both the node list and the Map view (see Figure 4). You can select the node by clicking on its title in the node list.

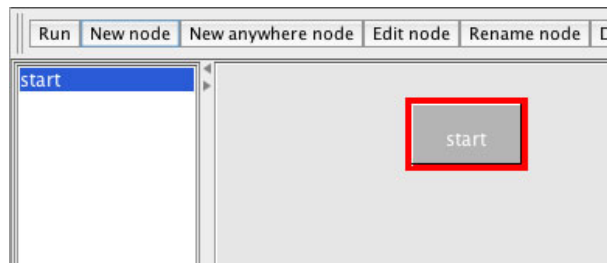


Figure 4: *Our first node showing in the Node list and Map view.*

You can also drag the node around in the Map view by selecting the node and then dragging it. Once we have created several nodes and links between them, it can be useful to position them in the Map view so that we can easily see the relationship between them. However, the position of the nodes in the Map view does not change their behaviour in any way.

4 Adding text to a node

Now that we have our first text node, we need to go in and enter some text.

1. Choose the title of our node, start, in the node list, and then click on the Edit node button in the Toolbar. This will open the editing view for the node, as shown in Figure 5.

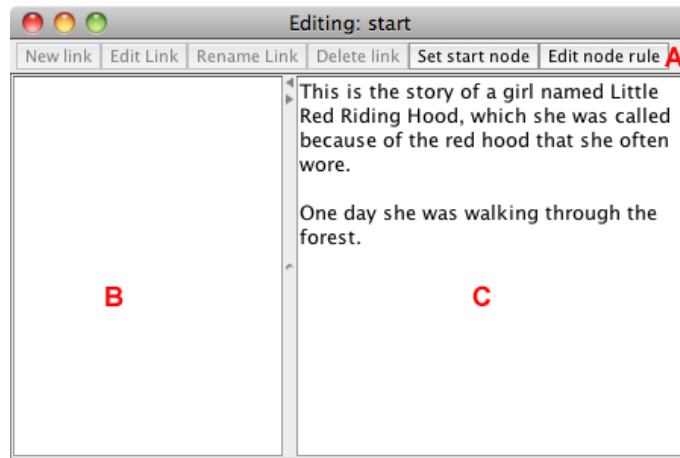


Figure 5: *The editing view of a text node.*

The Editing view lets you enter the text for a node. It also lets you add, edit and delete links. The Editing view consists of the Toolbar (A), the Link list (B), and the Content editor (C).

Note that the name of the node that you are currently editing is shown in the title bar of the Editing view.

2. Now enter the text for the first fragment of our story in the Content editor:
This is the story of a girl named Little Red Riding Hood, which she was called because of the red hood that she often wore.
One day she was walking through the forest.
3. We want this node to be the start of our story, so click on the Set start node button to set this node as the start node. Every story must have a start node. The start node is indicated by the words “(start node)” shown after the node name in both the Node list and the Map view.
4. Once you’ve finished entering this text, close the editing view by clicking on the close button. This will take you back to the Map view.

5 Creating links between nodes

We will now create several other nodes, to represent other fragments of our hypertext story, and we’ll create links between them.

5.1 Creating the other nodes

1. Create another text node, and name it “forest”.

2. Enter the following text in the node:

In the forest, Red came across a young man with a nasty smile.

“Where are you going, little girl?” he asked.

“I’m off to see my sick granny,” she said.

Well, you can probably guess what happened next.

This node represents the main conflict of the story.

3. Create a third node, and name it “end”.

4. Enter the following text in the node:

**** The End ****

back to start

This node represents the end of our story.

You should have three nodes: “start”, “forest”, and “end”, as in Figure 6.

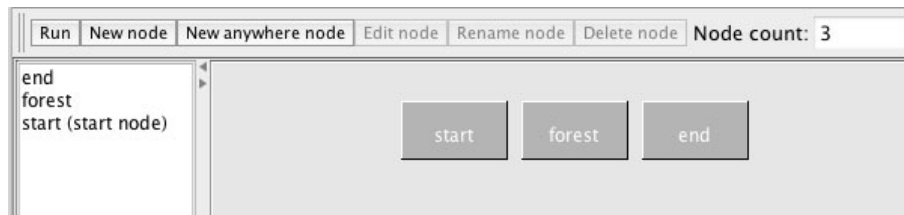


Figure 6: *Three text nodes.*

5.2 Creating a link

Now we want the user to be able to move from one text node to another, as they make decisions about how to read the story. To do this, we need to create links between the nodes.

Links in HypeDyn can be attached to a specific piece of text. When the user clicks on a link, one or more rules are triggered. These rules are where the author can specify what will happen when the link is clicked. For example, in our story, we want the user to be able to click on “forest” in the “start” node to jump to the “forest” node. This lets the user progress through the main story path.

1. Select the “start” node in the Node list, and then click on the Edit node button.
2. Now select the word “forest”. We’ll attach our link here.
3. Click on the New link button in the Editing view. A New link dialogue box will appear, as shown in Figure 7.

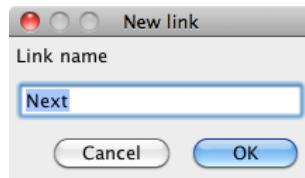


Figure 7: *Creating a link.*

4. Give the link a name, such as “Next”, and click OK.
5. You will now see the “Edit link” dialogue (see Figure 8). This is where we can specify the rules for the link.

The “Edit link” dialogue consists of a list of rules, each of which will be triggered when the user clicks on the link. For now, we will create a single rule, which will jump the story to the “forest” node.

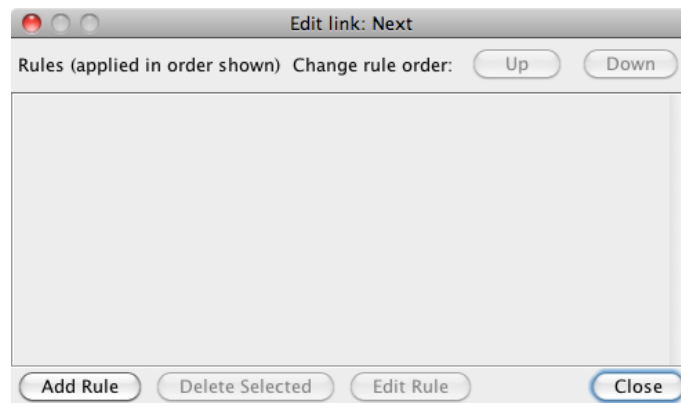


Figure 8: *The “Edit link” dialogue.*

6. Click on the Add Rule button at the bottom left of the “Edit link” dialogue. A rule, named “new rule”, will be added to the list of rules (see Figure 9).

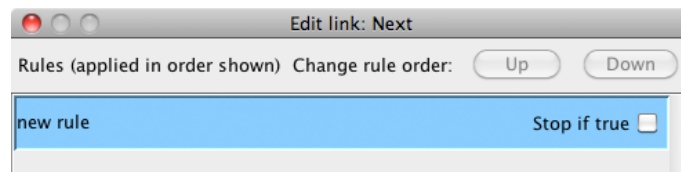


Figure 9: *Adding a rule to the “Edit link” dialogue.*

7. The new rule is initially selected. Click on the Edit rule button to edit the rule. You will now see the “Edit rule” dialogue (see Figure 10).

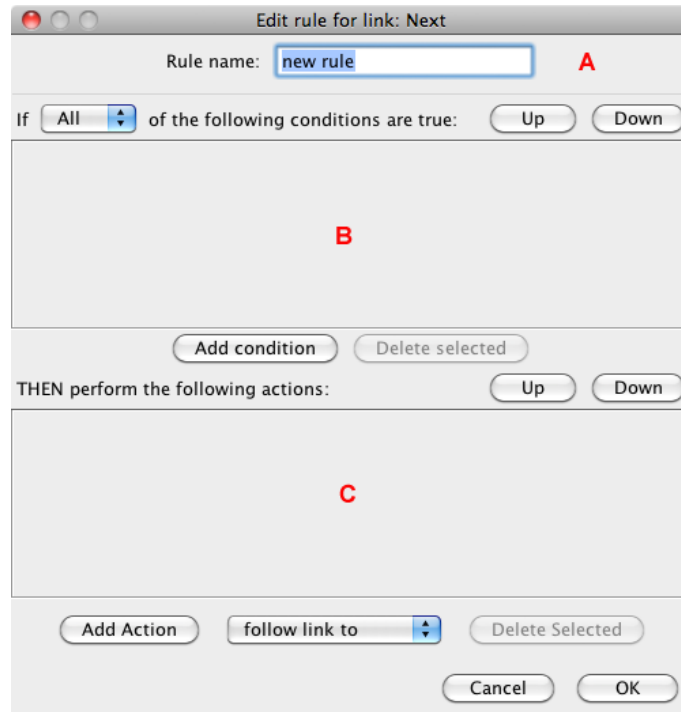


Figure 10: The “Edit rule” dialogue.

The “Edit rule” dialogue consists of three sections: the *rule name* (A) which allows you to edit the name of the rule, the *conditions* (B) listing a series of conditions which specify when the rule can be triggered, and the *actions* section (C) which is the set of actions that are performed when the conditions are satisfied. We will return to the conditions below.

8. First edit the name of the rule, changing it to “Next”. This is the label which will be placed on the link when it is displayed in the Map view.
9. Next, create the action. Select “follow link to” in the pull-down menu beside the “Add Action” button, and click on the “Add Action” button. An action will be added to the action list.
10. In the new action, use the pull-down menu to choose the “forest” node. This will create a link from the text “forest” in the “start” node to the “forest” text node (see Figure 11).
11. Now click OK in the “Edit rule” dialogue, and “Close” in the “Edit link” dialogue.

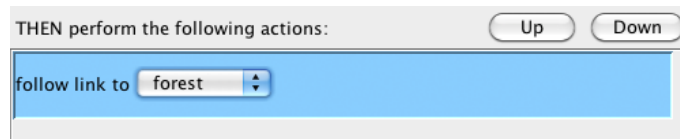


Figure 11: *Creating the rule to follow a link from “start” to “forest”.*

We have now created a link from the “start” node to the “forest”. You should be able to see this link in the Link list at the top of the Editing view, and in the Map view, as shown in Figure 12.

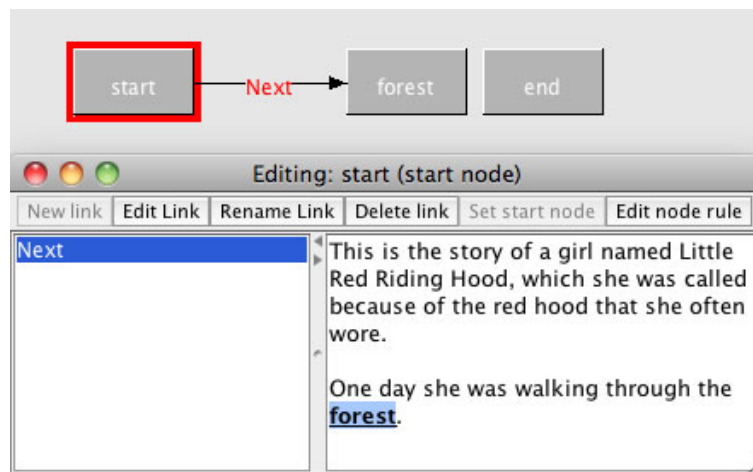


Figure 12: *The link from the “start” to the “forest”.*

Notice that links show up as bold, underlined text in the content of the text node.

5.3 Testing the link

Now that we’ve created the link, we need to test that it works.

To switch from editing to reading the text nodes, we need to use the Run button in the Toolbar in the Map view.

1. Close the “start” node if its still open.
2. Now click on the Run button. HypeDyn will open the Reader in a web browser window, showing the contents of the text node. You are now “reading” the story, instead of editing it, as shown in Figure 13.

In the Reader, you can click on links, and you’ll be taken to the text node that the link leads to.



Figure 13: *Reading a text node.*

3. Click on the word “forest” to follow the link.
4. The Reader will change to show the contents of the “forest” node.

You can finish reading the story by closing the Reader window.

5.4 Creating more links

Now that we have one link from the “start” to the “forest”, we should link in our third node, “end”. This will allow the user to move forward along the main story path from the “forest” node to the “end” node, the end of the story.

1. Open the “forest” text node.
2. Select the word “next”.
3. Click on the New link button.
4. Name this link “to the end”.
5. Create a rule, and edit the rule. Name the rule “to the end”.
6. Now add a “follow link to” action, and select the “end” node in the pull-down menu.
7. Click on the Ok button, and close the link editor.

You now have a link to the “end” text node. Test this out by running the story. You should be able to click on the “forest” link to move to the “forest” node, and then click on “next” to move to the “end” node.

Now we’ll create one more link, this time from the “end” node back to the “start”, so that the reader can re-read our story if they want.

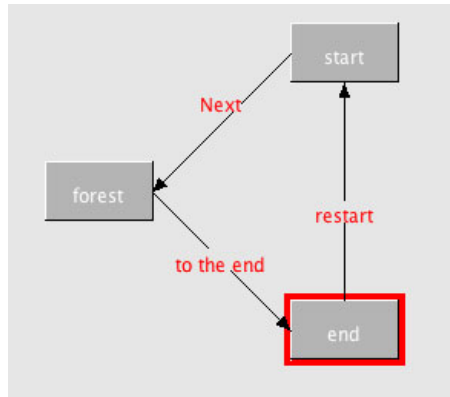


Figure 14: *Three nodes with a loop back to the start.*

8. Open the “end” node.
9. Select the words “back to start”.
10. Click on the New link button.
11. Name this link “restart”.
12. Create a rule, and edit the rule. Name the rule “restart”.
13. Now add a “follow link to” action, and select the “start” node from the pull-down menu.
14. Click on the Ok button, and close the link editor.

We now have a loop back to the start. At this point, your Map view should look as shown in Figure 14.

6 Creating rules with conditions

The next thing we want to do is create a rule with conditions. This is a rule that can only be triggered if a certain condition, such as the user having visited a specific node, is satisfied.

We will create a node, named “Hood details”, which can be reached after the reader has read the entire story. This node will reveal additional information that will (hopefully) change the reader’s experience of the story, as the reader will now know that there is something unusual, and perhaps slightly sinister, about Red’s hood.

To do this, we will use a feature in HypeDyn called a “condition” to only allow the reader to see the “Hood details” node after they have seen the “end” node. This allows us, as the author of the story, to ensure that the reader

has gone through the main story path at least once. We want the reader to experience the original story before seeing the new information, so that the new information can more effectively impact their previous mental model of the events of the story.

6.1 Creating the node

First we'll create a new node to contain the information about the hood.

1. Create a new node, and name it “Hood details”.
2. Enter the following text in the text node:
Her hood was a magic garment, given to her by her grandmother. It could kill anyone who tried to harm the wearer.
But not immediately. And in a most painful manner.
Meanwhile, in the forest...
3. Next, select the word “forest” in the text you just entered (in the “Hood details” node). Create a link on this word, and name the link “continue”. Add a rule to the link, and name it “continue”. Give the rule a “follow link to” action that goes to the “forest” node.

This link leads the reader back to the main story path, thus allowing the reader to continue reading the story after having read the additional details in this node.
4. Now create a link from the “start” node to the “Hood details” node. Edit the “start” node and select the text “red hood”. Create a link with a “follow link to” action that goes from this text to the “Hood details” node, and name the link and its rule “more about the hood”.

At this point, you should have the nodes as shown in Figure 15.

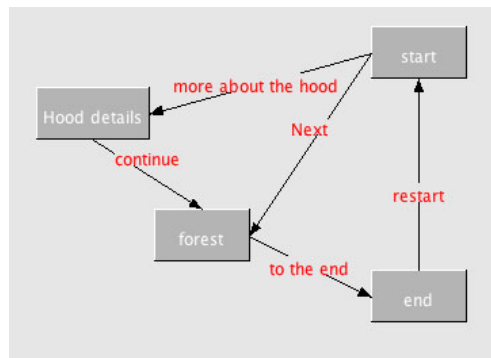


Figure 15: Setting up the “Hood details” node.

6.2 Adding the condition

Now we want to make it so that the reader can only follow the “more about the hood” link if they’ve already read the “end” node. To do this, we need to add a condition to the rule on the link between the “start” node and the “Hood details” node.

1. Open the “start” text node, and select the link “more about the hood”.
2. Now click on the Edit link button. The Edit link dialogue will appear.
3. Select the “more about the hood” rule, and click on the “Edit Rule” button. The Edit rule dialogue will appear.

The Edit rule dialogue lets us create conditions which must be true for the rule’s actions to be performed. If there are multiple conditions, then you can choose whether the actions will be performed when All or when Any of the conditions are true, by using the Any/All pull-down menu. For now, leave this pull-down menu at the default, All.

4. Click on the Add condition button. This will create a new condition which must be satisfied for the actions to be performed, as shown in Figure 16.



Figure 16: *Adding a condition.*

5. We want the action “follow link to Hood details” only to be performed if the reader has gone through the entire story at least once. This means they have to have visited the “end” node. Use the pull-down menus so that our condition reads “Node end Visited”. (Your condition should look like Figure 17.) Then click Ok, and close the Edit link dialogue.

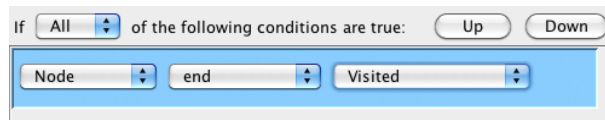


Figure 17: *Condition for the “Hood details”.*

6. Close the “start” text node.

6.3 Testing the rule

Now we will test out the rule.

1. In the main window, click on the Run button.
2. In the Reader you should see that the text “red hood” is bold, but not underlined. This indicates a link which cannot currently be followed, as shown in Figure 18.



Figure 18: *Start node with a link that cannot be followed.*

3. Now read through the story once, by clicking on “forest” in the “start” node, and then “next” in the “forest” node.
4. Finally, read back to the start by clicking on the “back to start” link in the “end” node.
5. This time, you should see that the “red hood” text is a normal, followable link. Click on the link to go to the “Hood details” node.
6. If you want to read the story again as if you’ve never read it, you need to clear the Reader’s history of which nodes have been visited.

In the Reader, click on the Restart. The Reader will go to the node which we set as the start node, “start”, and will clear its history.

7. You should now see that the “red hood” text is once again an unfollowable link.

We have now created a simple story, with links and conditions.

7 Reading the story

Once you have created and saved your story, you can export your story so that it can be read by other people. Choose File->Export for web... and specify the name of the folder where you want to export your story. HypeDyn will create a folder with that name, and save your story there as a web page. The folder contains a number of supporting files, plus an “index.html” file. Keep all these files together.

To read your exported story, open the “index.html” file in a web browser. This is exactly the same as running the Reader from within the editor.

8 Conclusion

In this tutorial, we have created a simple hypertext fiction. Our story has a main path, which tells a simplified version of the traditional folk tale “Little Red Riding Hood”. Our story also has an alternative side path, which gives the reader more information about Red’s hood. This alternative side path can only be read once the reader has visited the “end” node at the end of the main story path. Although this is a simple story, it demonstrates all the capabilities of HypeDyn, capabilities which are sufficient to create a more complex hypertext fiction.

Please see tutorial 2 for more advanced features: links with multiple rules, conditional text, and anywhere nodes.