

# Virtual Tutoring Previewer Tutorial

---

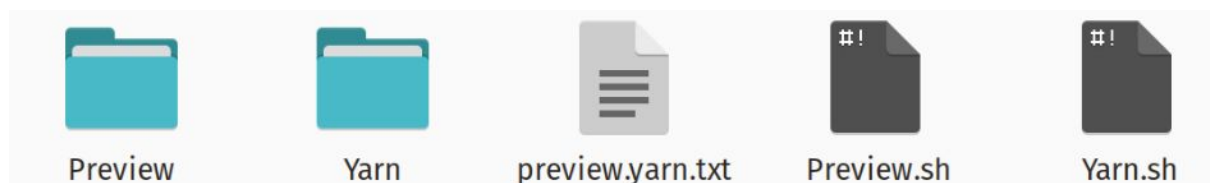
In this document we present a small tutorial on how to use the **Virtual Tutoring Previewer** (VTP). First is presented an overview of the system followed by a step-by-step tutorial on how to use the different components.

## Overview

In this section, we will present a brief overview of all the components present within the **VTP** application. The remaining sections will be dedicated to practical tutorials on how to use the application to create and visualize dialogues.

## Directory Structure

Accompanying this document are 3 important files: *Yarn*, *preview.yarn.txt* and *Preview*. The usage of these files will allow you to create/modify dialogues, test them and save them for later use.



**Fig. 1.** An overview of the Previewer directory.

## Yarn

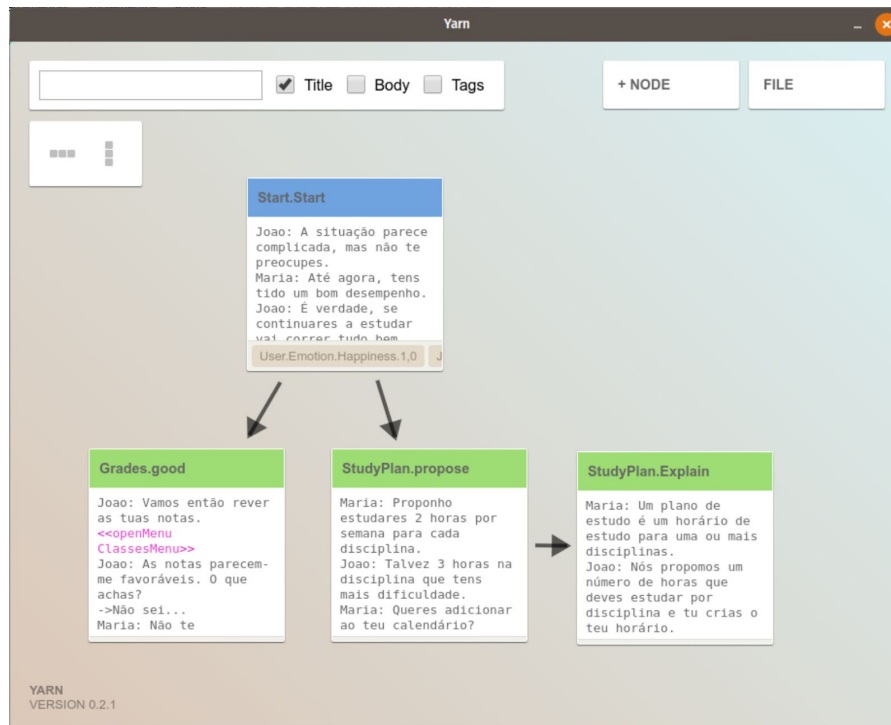
By double-clicking the file **Yarn** (not the folder with the same name) an application is launched<sup>1</sup>. This application can be used to perform various tasks.

It allows the creation of new dialogs from scratch, by giving the user the ability to create new nodes and modify the options inside them. It also lets the user save those dialogues, as well as open already existing ones, via the “File” menu on the top right corner.

In figure 2, you can see what we are presented with when we use the application to open the **preview.yarn.txt** file.

---

<sup>1</sup> In MacOS the application may be tagged as unsecure. To open it the first time press **Control + Click > Open**, this will allow you to open the application.



**Fig. 2.** The preview.yarn.text file, opened in the Yarn application.

## preview.yarn.txt

After creating a dialogue in **Yarn**, we can save it in a file. **preview.yarn.txt** is an example of a dialogue. We'll use this file to better explain the use of VTP.

## Preview

Double-clicking this file will open the Previewer<sup>2</sup>. This application will read **Yarn** files, containing dialogues, and allow the user to test **Virtual Tutoring** application running the created dialogues. This application allows dialogue reading, option selection and command interpretation, more will be explained later.

<sup>2</sup> In MacOS the application may be tagged as unsecure. To open it the first time press **Control + Click > Open**, this will allow you to open the application.



**Fig. 3.** The Previewer application.

## Step-by-Step Tutorials

In this section, we will present three tutorials.

The first tutorial helps you explore the Yarn and Previewer platforms. The second tutorial will guide you through building a new dialogue from scratch using Yarn, and viewing it in the Previewer. The last tutorial will assist you in further enhancing your dialogue, by issuing commands to the Tutors.

### Exploratory Tutorial

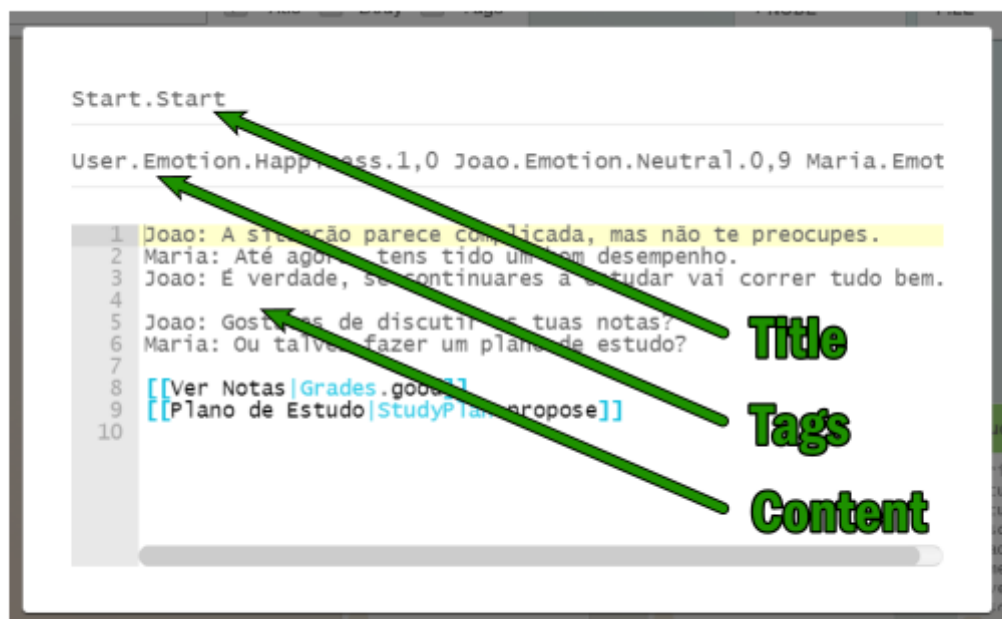
- 1) Double-click on **Yarn** to open it.
- 2) On the top right side, click on **File > Open...**
- 3) Select the **preview.yarn.txt**.
- 4) Explore the dialogue.
  - a) You can drag nodes to move them around

- b) You can also open\edit nodes by double-clicking them. This will cause a window to appear. Click anywhere outside of that window to return to the dialogue overview.
- 5) Double-click on **Preview** to open the application.
  - 6) On the field **Select Yarn File** press the File Chooser Button and select the **preview.yarn.txt**. Then press the play arrow when the file is selected.
  - 7) On the field **Start Node Name** type **Start.Start** and press the play arrow button.
  - 8) Experience the dialogue and select the options when presented.

## Basic Tutorial

In this tutorial we'll modify the **preview.yarn.txt** dialogue, by adding a new node to it, and then running it using the **Previewer**. It will be a dialogue that welcomes the user

- 1) Double-click on **Yarn** to open it.
- 2) Click on **File > Open...** and select **preview.yarn.txt**.
- 3) Create a new node by pressing **+ Node**.
- 4) Click and drag to move it.
- 5) Double-click on the node to edit it. The node has three important fields: Title, Tags and Content; (see figure below).



- 6) On the first line, change the node title to **Start.Hello**.

- 7) Let's now add tags defining what the tutors are feeling. Let's make João be with a neutral expression and Maria feel slightly happy. On the second line add the following tags:

a) **Joao.Emotion.Neutral.0,9 Maria.Emotion.Happiness.0,5**

- 8) We'll now create the dialogue the tutors will discuss. On the content text box, replace the text **Empty Text** with a welcoming dialogue:

Maria: Hello there!  
Joao: Long time no see!  
Maria: Do you mind if we talk about a Study Plan?

a) Note that, in the content text box, empty line are ignored.

- 9) Now let's add options to navigate to other options:

[[Of Course!|StudyPlan.Explain]]  
[[Could you repeat that?|Start.Hello]]

a) The first option leads to the node that explains what is a study plan.

b) The second option repeats this node from the start.

- 10) Return to the dialogue overview by clicking in the darkened part of the background. You will see an arrow leaving the node we created to the node named **StudyPlan.Explain**.

- 11) Let's save the changes by clicking **File > Save**.

- 12) With the dialogue created let's test it using **Preview**. Double-click on **Preview** to open it.

- 13) On the field **Select Yarn File** press the File Chooser Button and select the **preview.yarn.txt**. Then press the play arrow when the file is selected.

- 14) On the field **Start Node Name** type **Start.Hello** and press the play arrow button.

- 15) Experience the dialogue and select the options when presented.

## Advanced Tutorial

In this tutorial, we will build upon the **preview.yarn.txt** dialog we used in the previous tutorial, by adding some of the available commands supported by VTP. After we're finished, we will run it using the **Preview**.

- 1) Double-click on **Yarn** to open it.
- 2) Click on **File > Open...** and select **preview.yarn.txt**.
- 3) Double-click on the node previously created (Start.Hello) to edit it.

- 4) Now let's create a different option:

```
[[I don't want to.|Negative.Response]]
```

- 5) Now we need to create the corresponding node. Create a new node and open it.

- 6) Change the node title to **Negative.Response**.

- 7) If that option was to be chosen, the tutor emotion should change. For that, we have two solutions:

- a) Add different tags for this node.
- b) Add a command to change the emotion.

We will be using a command this time, since it can be used multiple times within the same node, therefore allowing multiple emotion changes.

- 8) Let's make Maria afraid and João sad, by adding the following commands:

```
<<Feel Maria Fear 0.4 5 None>>  
<<Feel Joao Sadness 1.0 5 None>>
```

- 9) With this, Maria mood will become afraid, as well as the corresponding background. Similarly, João will start to become sad. The last two parameters, control the duration of the background transition and the icon that should appear in the background. We can neglect them for now.

- 10) Now add the dialogue they should present:

```
Maria: Why not? Is everything okay?  
Joao: You should be more concerned about your studies!
```

- 11) Let us make Maria show immediate agreement with Joao's statement by nodding for 1 second. After Joao's line, add the following commands:

```
<<Nod Maria Start>>  
<<wait 1>>  
<<Nod Maria End>>
```

- 12) Let's add one last line of dialogue, this time with Maria in a Neutral mood:

```
<<Feel Maria Neutral 1 3 None>>  
  
Maria: Well, when you feel like talking, come back to us.
```

- 13) We can now use the wait command to wait a few seconds after the dialogue, and finally use the exit command to exit the application.

```
<<wait 3>>  
<<exit>>
```

- 14) With the dialogue created, save the changes and test it using **Preview**.
- 15) On the field **Select Yarn File** press the File Chooser Button and select the **preview.yarn.txt**. Then press the play arrow when the file is selected.
- 16) Experience the dialogue and select the new option when presented. If you would like to learn about other commands, you can read the documentation and experiment at will.