SceneMaker Authoring Tool

Tutorial Version 1.0
# TABLE OF CONTENTS

## TUTORIAL 1
- How to create a simple scene including an animation and speech output

## TUTORIAL 2
- How to connect several Nodes in a row

## TUTORIAL 3
- How to create a Timeout Edge
- Defining variables in order to use the Conditional Edge
- Using the Conditional Edge
- Various Conditional Edges

## TUTORIAL 4
- Probability Edges
- Introducing Supernode

## TUTORIAL 5
- Playing two Nodes synchronously

## TUTORIAL 6
- Introducing User-Input

## TUTORIAL 7
- How to play several scenes simultaneously

## FINAL PROJECT
- How to connect all tutorial

## ADVANCED TUTORIAL
- Using History

---

Imagine: You want to create a scene in which a woman called Gabi introduces herself and moves her hands. This can be realised in several minutes and is easier than you think.
How to create a simple scene including an animation and speech output

TUTORIAL 1
First of all drag the Basic Node (an element appearing in the left Tool Box) into the working area the so called Scene Flow Editor.
Make a right-click on the Basic Node and click “Set Start”. Since now the SceneMaker knows where the beginning of the chain is.
The red arrow marks the starting point of your scene.
As an example please type in the text into the Scene Script Editor. Now we have already created a little scene.

Scene_en: WelcomeScene
G: Hello [anim id=10]. My name is Gabi. I will show you round [anim id=23]. We are here in a beergarden at a class reunion.
G: Now we can take a look around who has already arrived.
Right now push the “abc-button” in order to check the syntax of your text. After that, drag the scene from the Tool Box and drop it on the Node.
In addition to that it is always helpful to give your scene an individual name. Just type in the new name of your scene into the space in the right upper corner. You can also see that the “PlaySceneGroup” appears in the right side now.
How to connect several Nodes in a row.

TUTORIAL 2
Right now we want to create a scene that is more complex than before. The scene of the first part of the tutorial will be our basis.
Delete the starting point of our first scene by doing a right-click on the Basic Node “Welcome” and choose “Unset Start”.
Add a Basic Node as always by dragging it into the working area. Name it “HeidiIntro”.

Hello [anim id=10]. My name is Gabi, I will show you round [anim id=23]. We are here in a beer garden at a class reunion. Now we can take a look around who has already arrived.
Create a new starting point for this scene by making a right-click on the Basic Node and select “Set Start”. 
The red arrow shows the starting point of your scene.
As an example please add the following text into the Scene Script Editor.

Scene_en: HeidiIntro
G:Hey Heidi [anim id=13]. Nice that you could arrange coming here. I am glad to see you [anim id=11] Come over here!
Then drag the scene from the Tool Box and drop it on the Node.
Add another Basic Node in order to play several scenes in a row. As always: Drag the Basic Node from the Tool Box into the working area.
Rename the Basic Node into “HeidiComing”.
Add the new text into the Scene Script Editor.

Scene_en: HeidiComing
H:[goto ent=G].
Connect those two scenes with an Epsilon Edge by dropping the Edge from the Tool Box on the left side into the working area between the two scenes.
The arrow should look like this afterwards.
Create a new scene and name it “HeidiTalking”. The new scene will appear in the Tool Box on the left side.
Right now it would be interesting to let Heidi talk with Gabi. Therefore we need another Basic Node that must be dropped again from the Tool Box into the working area.
Klick “enter” and the Basic Node will be renamed.
Connect the two scenes “HeidiComing” and “HeidiTalking” with an Epsilon Edge. All you have to do is drag the Edge from the Tool Box into the working Area between those two scenes. Afterwards the three created scenes can be played in a row.
Now it would be interesting to listen to what Heidi says. Therefore add the text (as an example) into the Scene Script Editor.

Scene_en:
HeidiTalking
H:Hey Gabi.Nice to see you! I am working here as a waitress. So I am very busy at the moment. But we can talk in two hours [anim id=19] when I have finished my work.
H:Oh Mike and Tom are standing overthere if you might not have noticed them yet. See you later [anim id=14].
Then drag the scene from the Tool Box and drop it on the Node.
The story would definitely be incomplete if Heidi isn’t going away. Therefore we need to add another Basic Node into the working area. As always: Drag the Basic Node from the left side into the working area.
Rename the Basic Node into “HeidiLeaving” by typing the new name of the scene into the box in the right upper corner. Click “enter” and the new name of the scene will appear in the Basic Node.
Connect the third with the fourth scene by dropping an Epsilon Edge from the Tool Box into the working area between both scenes.
For an appropriate ending please add the text (as an example) into the Scene Script Editor.

Scene_en HeidiLeaving
G: Bye, bye.
H: [goto dst=bar].
Then drag the scene from the Tool Box and drop it on the Node.
How to create a Timeout Edge.
Defining variables in order to use the Conditional Edge.
Using the Conditional Edge.
Various Conditional Edges.

TUTORIAL 3
Create a new Basic Node as always: Drag the Node from the Tool Box into the working area. Furthermore delete the starting point which you have created at the beginning by making a right-click on the Basic Node “HeidiIntro” and choose “Unset Start”.

To let Gabi getting in contact with the men’s world right-click on the Basic Node “N6”, set it as a Start Node and then rename it to “Gabi and Boys”. Afterwards add a New scene into the Scene Script Editor to let Gabi (G) go to Tom (T).

Scene_en : GabiandTom
G: [goto ent=T].
Add a new Basic Node into the working area as you already did before and rename it to “TalkingTom”. Connect the two Nodes with an Epsilon Edge and add the text into the Scene Script Editor. Afterwards drag the new scene „TalkingTom“ from the Tool Box into the Working Area.

Scene_en: TalkingTom
G: Hey Tom! [anim id=23] We have not seen each other for a long time! What is going on! How are you?
T: [anim id=32] I am fine thank you and you? I was on vacation a week ago.
G: Oh, really [anim id=35]. How was it? Where have you been?
T: [anim id=34] I was in Greece and it was wonderful.
Add a new Basic Node “Mike Talking”. 
Use the Timeout Edge to connect the Node “TalkingTom” with “TalkingMike”.

After that a new window appears.
Change the Timeout value from 1000ms to 20000ms (1000ms = 1 second).
The timer starts when the brown Node starts with the scene. After 20 seconds the next Node will be started.
Create the Scene "TalkingMike" including the text and add it to the Scene Script Editor.

Scene_en : TalkingMike
M: Hey you turtledoves. I am here as well! Let’s have a drink together. Oh my god I am drunk.
Add some animations to the Scene in order to make the conversation less monotonous. Lean back and watch the scene.
Since know the tutorial gets in contact with interactive storytelling.
Add a new Basic Node “DrunkenMike” and use the Epsilon Edge to connect it. Write 3 new Scenes “DrunkenMike” into the Scene Script Editor and use different sentences in order to let Mike say randomly 3 different sentences. After that drag the scene “DrunkenMike” from the Tool Box on the Node.

Scene_en : DrunkenMike
M:I am so drunk.
Scene_en : DrunkenMike
M:Wow this drink is nice.
Scene_en : DrunkenMike
M:Oh my god this is some good beer
Now connect the Node “DrunkenMike” with itself using the Epsilon Edge to create an infinite loop.
But this infinite loop is not realistic for story writers. That’s why we want to show something that is a little bit more realistic. Delete the Epsilon Edge by right-clicking on it and select „delete“. 
To continue with our story it is time to introduce the Conditional Edge. First of all some variables are needed. Therefore click on the “+” button on the right side of your screen below the “Edit Variable Definitions”.
Set the type to „Int“.
Name the variable “drunkNumber”. Last but not least you have to set a value by clicking on the three dots.
A new pop up appears. Set the value to “0”.
The variable now has the value 0. Every operation that deals with that variable relies on that value unless you give the command to change it. It should now look like this.
The variable is mentioned here.
Now it is time to use the Conditional Edge. Connect the “DrunkenMike” Node with itself like you did previously. A new window will appear.
You have to set the conditional expression(s) here. In this case the Node will be played as long as the drunkNumber is smaller than three. Confirm with "OK".
The next step is to set the “Command Executions”. Make sure that you have chosen the right Node in this case the yellow one “DrunkenMike” --> click on the “+”.
And type in your command.

drunkNumber = drunkNumber + 1
That means that the variable “drunkNumber” increases with every repetition by one. The Node will execute following commands now.
Add a last Node.
Connect it with the Epsilon Edge. Add the scene with the text you can see on the right side and Mike will leave the conversation and goes away.

Scene_en : MikeLeaving
G:I think you have drunk too much.[anim id=16] Maybe you should go home and sober up.
M:What?No way!I think I will take another drink![anim id=18] Oh, I think now I have to leave.
M:Bye [anim id=15].
M:[goto dst=home].
Now it’s still not realistic enough. To make it more realistic an additive Conditional Edge is needed. Add a new scene “MikeEating”
Also add a new variable “stamina” like you did before. Set the value to 2.
Click on the “+” to add another Command Execution
Type in the following text. This time the variable decreases by every repetition by one.

stamina = stamina - 1
The command will be executed now.
Select “MikeEating” and add a Command Execution.
That impacts that the stamina will be set to two. Now modify the Conditional Edge.
Change the “drunkNumber” from “3” to “5” (this is necessary to let the system know that there is another variable integrated and limits the length of the minigame). && means that both conditions have to be fulfilled to activate the Node. Also type in that the stamina has to be > 0.

(drunkNumber < 5) && (stamina > 0)
Connect the Conditional Node with “MikeEating” by using another Conditional Edge.
It should look like this now.
That means that if the “drunkNumber” is < 5 and the stamina is <= 0 the Node “MikeEating” will be run.

(drunkNumber < 5) && (stamina <= 0)
Your working area should look like this now.
Last step: Connect the “MikeEating” Node and the Conditional Node “DrunkenMike” with an Epsilon Edge. Press play and enjoy the scene.
Probability Edges.
Introducing Supernodes.

TUTORIAL 4
Unset the Starting Node from tutorial 3. Set a new Basic Node as start Node, add a new Scene with text (“GabiTalking”).

Scene_en : GabiTalking
G: It is better if he goes home.
G: Hey Tom! Do you remember Susan?[anim id=38].
Add three Basic Nodes “Yes”, “No”, “Help”. 
Scene_en : Yes
T: Oh yes of course. [anim id=22] I know her since we were 10 years old. Let us go over.
Scene_en : No
T: No I do not know her. [anim id=26] What was her name. does not matter, let us go over for a talk.
Scene_en : Help
T: Not really. [anim id=28] Can you give me a hint? I really do not have a clue. But maybe she could tell me. Let us go over.
Connect the “GabiTalking” Node with the “Yes” Node by using a Probability Edge.
A new window appears. The number in red shows which Edge is activated at the moment. This number also states the percentage probability the following Node will be executed.
Add a second Probability Edge to connect “GabiTalking” with “No”.
The probabilities have to be separated so that the Rest is $= 0.$
For example like this.
The Button “Normalize” means that the size relations stay the same and will be converted to 100. The Button “Uniform” means that all Edges have the same possibility. The rest have to be adjusted manually.
Connect the third Node as you did before.
Set the percentages to something like this.
The probability percentages will be shown on the Edges.
Add a new Basic Node and connect the three previous Nodes with it by using an Epsilon Edge. The “GoSusan” scene makes Gabi and Tom go to Susan one after another.

Scene_en : GoSusan
G:[goto ent=S].
T:[goto ent=G].
Drag a Super Node in the working area.
Select the Nodes “GabiTalking”, “Yes”, “No”, “Help” by holding down the key shift and click right. Select “Copy Nodes”. 
Double-click on the Super Node and a new layer will appear.
Right-click and paste the copied Nodes.
The result should look like this now. Leave the layer by clicking on the arrow.
Add a new Basic Node and add the Scene “GoSusan” – the result in AAA should stay the same as it is in 4.1., but it is more understandable in SceneMaker.
Playing two Nodes synchronously.

TUTORIAL 5
Rebuild the following structure of the Nodes and the Edges. Set two Starting Nodes.
Add the scenes with the scripted text, that’s all the magic. The nodes will be played simultaneously.

Scene_en : Susantalk1
S: Oh, hello Gabi! Have you seen the girl over there?
Scene_en : Susantalk2
S: She looks like a topmodel!
Scene_en : Susantalk3
S: Oh, sure I know her. She has changed. She looks very nice. Hello User!
Scene_en : Gabitalk1
G: Yes, I take her around the beergarden!
Scene_en : Gabitalk2
G: You do not you recognize her? It is User, our old classmate.
Introducing User-Input

TUTORIAL 6
In this tutorial we would like to show you how to make the story interactive by using the user interface for text input. At first we build our scene with text where Susan asks the user how he feels.
Now we want the user to have the possibility to enter an (user generated) answer via text input. So we add a new Node where we enable the user interface for text input. Therefore we need to create some type definitions.
Set the flavour to ‘Struct’. As name set DAct and click on ‘Add’ to define the members.
At first define the member type by entering the Name “typ”. After that select String for the (data)type and confirm with OK. But there need to be two more members (txt, utt) specified by using the same procedure.
The result should look like this.
Now it is time to set the required functions for this tutorial: to activate the user interface for text input click on the “+” at the Edit Function Definitions.
Insert the following settings to open the user interface for text input.
Of course the user interface for text input has to be closed after using it. This is necessary for the functionality. Therefore it is useful to integrate this function right now in order not to forget it later.
Now the functions appear in the Tool Box below the folder functions.
Add the function “enableGUI” to the Node “enable” by using drag and drop.
It is indispensable to create another variable (in this case the dAct) to ensure that the user input operates based on string value.
Attention should be paid to the curved brackets.

{typ = "", txt = "", utt = ""}
Another variable is needed to request if the input has been done or not. This occurs with the aid of a boolean value.
Now create two new Nodes (wait&output)...

S: She looks like a topmodel!

Scene_en: Susantalk3
S: Oh, sure I know her. She has changed. She looks very nice. Hello User! How are you?

Scene_en: Gabtalk1
G: Yes, I take her around the bessgarden.

Scene_en: Gabtalk2
G: Do you not you recognise her? It is User, our old classmate.
... and connect them with a Conditional Edge.
Type in the following condition and confirm. This means that input has occurred and now the value is set to true.

(InputOccurred == true)
The result should look like this.
In case that no user input has been made, the Node „wait“ waits for an adequate user input which can be realized by inserting a Timeout Edge.
Select the Node „Output“ and determine specific commands.
By using the following command set the value back to false because the user input has been made.

```
InputOccurred = false
```

InputOccurred = false
Moreover the GUI has to be disabled, so drag and drop the corresponding function from the Tool Box onto the Node (“Output”).
Now the scene is connected with the Node.
Now define the command to emit the entered text in the AAA.

```
PlaySceneGroup("Output", {text = dAct.utt})
```
In this case the scene Output contains „U:$text“. This is necessary to transfer the corresponding parameters that means the scene only includes what the user has entered.
Create three additional Nodes with the appropriated scenes (good, bad, askagain). Connect the source Node „Output“ with the target Nodes good and bad by using a Conditional Edge for each Node. Later the Node „askagain“ will be connected by using an Epsilon Edge.

Scene_en:
AskAgain
S:Whatever you say.
Scene_en:
ReactToGood
S:Nice I am happy for you.
Scene_en:
ReactToBad
S:I am so sorry, I hope you are doing better.
As mentioned above, connect the Conditional Edge to the Node “good” with the following condition. That means that if the entered text has been identified as a “goodmood” the Node (good) will be played.

(dAct.typ == "goodmood")
Same procedure for the bad Node with „badmood“

(dAct.typ == „badmood“)
Last but not least has to be clarified, what happens if the text is neither identified as „goodmood“ nor as „badmood“. For this purpose we connect the „output“ with the „askagain“ and finally with the starting Node „helloUser“. That means that the user will be asked continuously until he gives an evaluable answer. This means an answer that is defined in the spin rules (in templates.dll).
This is how the spin rules should look like after your modification.

Copy and paste the following command into your templates.dll in the folder"..\Soap\res\templates.tdl"

```xml
#mood
good not(Type())
  -> Type (value: "goodmood")
very good not(Type())
  -> Type (value: "goodmood")
fine not(Type())
  -> Type (value: "goodmood")
excellent not(Type())
  -> Type (value: "goodmood")
bad not(Type())
  -> Type (value: "badmood")
very bad not(Type())
  -> Type (value: "badmood")
terrible not(Type())
  -> Type (value: "badmood")
```

This is how the spin rules should look like after your modification.
How to play several scenes simultaneously.

TUTORIAL 7
At the end it is always helpful to introduce the Fork Edge. With that Edge many Nodes can be connected and played simultaneously. First of all, please create the Nodes and scenes like we did below.

Scene_en : Bye
G: I think we have seen enough for today. Good bye.
Scene_en : WaveGabi
G: [anim id=13].
Scene_en : WaveSusan
S: [anim id=13].
Scene_en : WaveHeidi
H: [anim id=13].
Scene_en : WaveTom
T: [anim id=13].
Scene_en : WaveUser
U: [anim id=13].
Scene_en : WaveMike
U: [anim id=13].
Afterwards please connect all the Nodes from the starting point with the Fork Edge.
Should look like this:
How to connect all tutorials.

FINAL PROJECT
In order to see the project without interruption we must connect the separate tutorials with each other. To do this, connect each end Node of the separate tutorials with the corresponding starting Node of the following tutorial by using the Epsilon Edge. But there are two exceptions.
Exceptional1: at the concurrently played Nodes the corresponding Fork Edge must be used. (Scene GotoSusan -> Susantlk1 and Gabitalk)
Exceptional 2: Both end Nodes ("good" and "bad") must be connected with the Fork Edge ("bye")
Using History

ADVANCED TUTORIAL
At first create a Super Node and a Basic Node with a test scene. Connect both with Epsilon Edges.
After that, enter the Super Node by double clicking on it and build three new Nodes with corresponding scenes (Intro, Answer and Explanation).
First of all, connect the History Node with the Node „Explanation“ due a Conditional Edge. In this case the condition checks if the Super Node(S1) and the Node (N5,Explanation) has been already played in the past. If this is true, you will only hear the Node Explanation and not the other Nodes (e.g. it doesn’t make sense that someone always introduces you to another person every time you are near the person)
There will be other advanced in the near future, there you will learn how the Interruptive Edge works and there are some more interesting things to work with. But now enjoy what you have learned and feel free to practice now on your own.