

Virtual Patient Encounter Script Structure

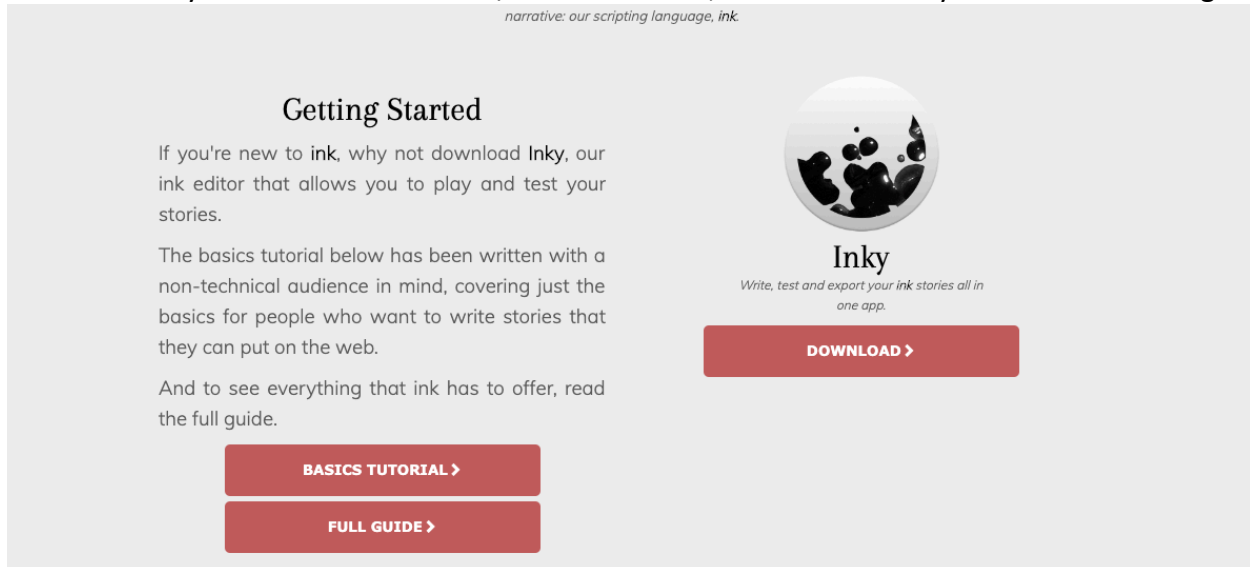
Table of Contents

<i>Inky Dialog Tool</i>	1
Opening an Encounter	2
Include Files	4
<i>Technical Content</i>	4
Hashtags	4
Functions	5
Variables	6
Setup	6
Speakers	6
Introductions.....	6
Topic Variables	6
Recap/Encounter Close.....	7
Sequencing.....	7
<i>Structure</i>	7
User Flow	7
Encounter Start	8
Introduction	9
Visit Purpose	9
Topic Hub	10
Social History.....	11
Medical History	11
Medications.....	11
Disease States	11
Diet & Exercise	11
Spacer.....	12
Encounter Close	12
Summary & Feedback	13
Hashtags and Variables	15

Virtual Patient Encounter Script Structure

In order to run or edit an encounter script, you will first need to download Inky. Inky is a narrative editing tool that we used to author all virtual patient encounters. *(Note: you can edit an encounter script in NotePad or TextEdit, but you are not able to test an encounter without either a nXhuman build, or Inky.)*

Download Inky from inklestudios.com/ink. Once there, scroll down and you see the following:



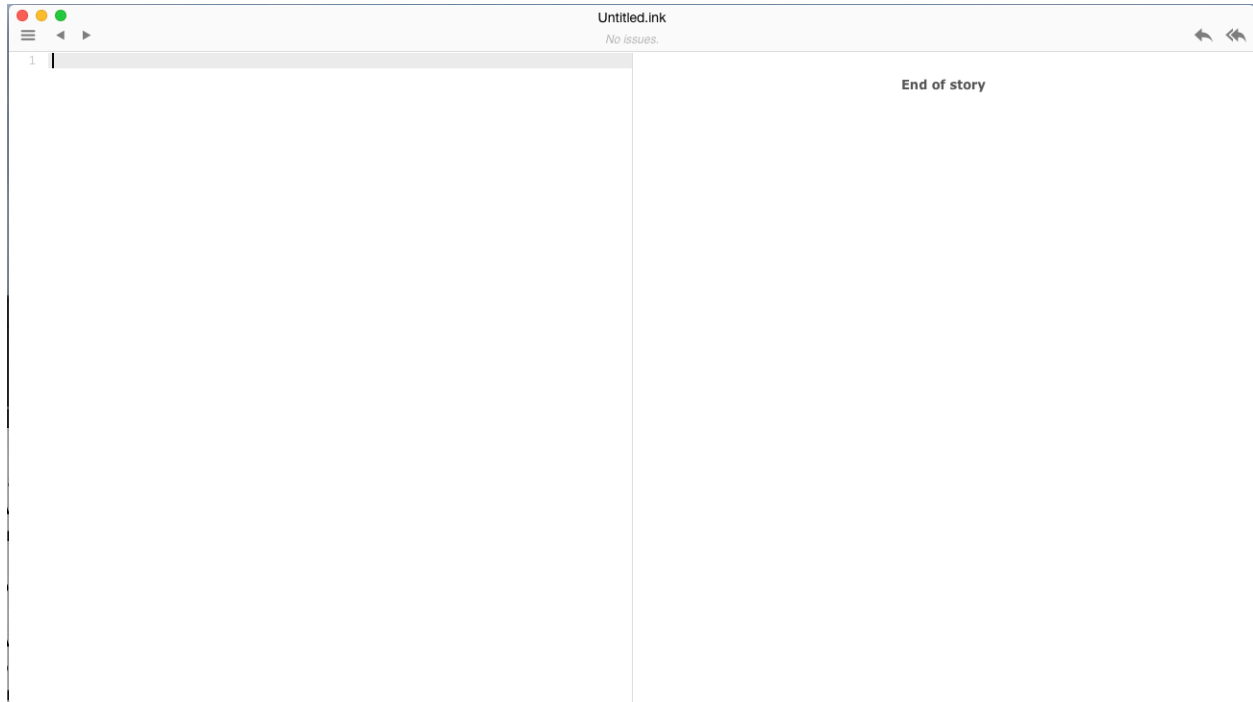
The screenshot shows the 'Getting Started' section of the Inky website. At the top, it says 'narrative: our scripting language, ink'. The main heading is 'Getting Started'. Below it, there is a paragraph: 'If you're new to ink, why not download Inky, our ink editor that allows you to play and test your stories.' This is followed by another paragraph: 'The basics tutorial below has been written with a non-technical audience in mind, covering just the basics for people who want to write stories that they can put on the web.' A third paragraph says: 'And to see everything that ink has to offer, read the full guide.' To the right of the text is a circular logo for Inky, which features a stylized black ink splat on a white background. Below the logo, the word 'Inky' is written in a bold, sans-serif font, followed by the tagline 'Write, test and export your ink stories all in one app.' Below the tagline is a large red button with the text 'DOWNLOAD >'. At the bottom of the page, there are two more red buttons: 'BASICS TUTORIAL >' and 'FULL GUIDE >'.

Click the download link and select the version for your system (Windows, Mac, etc.). If, after reading this guide, you feel ready to edit the scripts, you will also find a very helpful guide on writing within Inky.

Opening an Encounter

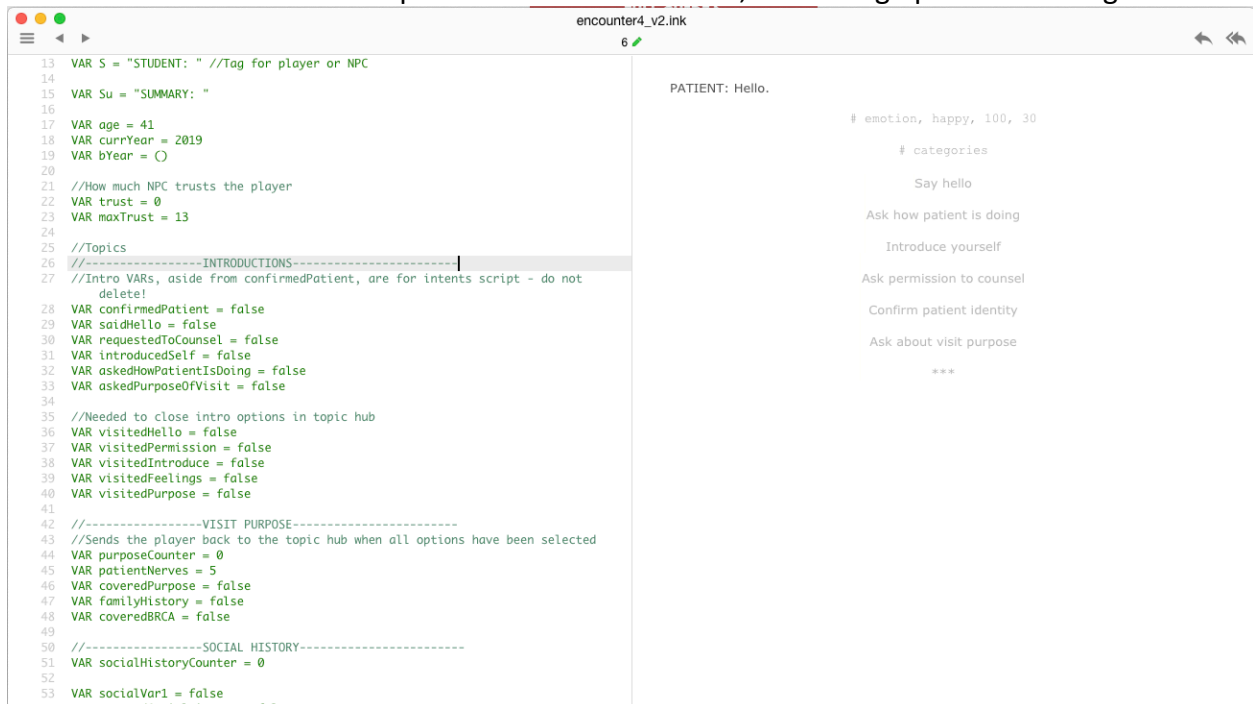
Now that you've downloaded Inky, you'll notice that upon launching the program, you'll be met with a blank page split into two columns.

Virtual Patient Encounter Script Structure



Navigate to File, and then click on Open. Select one of the nXhuman encounter scripts from where you have saved it on your computer.

Voila! You'll now see a full script in the left-hand column, and dialog options in the right.



Virtual Patient Encounter Script Structure

Unless you are planning to edit a script, you do not need to worry about the text in the left column.

When stepping through an encounter, you can use the single back arrow in the top right corner to reverse one choice, or click the double arrow to restart the entire encounter.

One final note – Inky confusingly enough writes in the ink language. This is why all encounters are saved as a .ink file. So remember, inky = program, ink = language/file type.

Include Files

Include files are small Inky scripts that are attached to full encounter files, mainly so we could reuse content. They are referenced within the main ink script but are not contained within the script itself.

(For a full guide on includes, check out the Ink/Inky tutorial on inklestudios.com/ink)

When opening the first three encounters, you will notice a ton of errors. This is because Inky does not create references to includes by default, and you will need to set this up.

In order to run encounters 1, 2, and 3, you will need to recreate the reference to the includes:

1. Within Inky, click on the hamburger menu icon in the top left corner.
2. At the bottom of the now open toolbar, click on the button reading “+ Add new include.” Inky will open a new, blank ink script. This is now attached to the main ink file.
3. Open `encounter_all_intro.ink` and `learning_objectives.ink` from where you have saved the files on your computer.
4. Copy the text from the left-hand column of one file and paste it into the new blank script.
5. Save, and repeat steps 2-4 for the other include file.

You should now be able to run the first three encounters in Inky without error.

As of July 2019, encounters 4 and 5 are completely standalone. Encounters 1, 2, and 3, however, need specific include files in order to run.

Technical Content

The following is an overview of an encounter’s ink code. This is meant to give you an understanding of our script functionality and is *not* a complete guide to writing in ink.

Hashtags

Virtual Patient Encounter Script Structure

Used to delineate specific content that Unity is then able to parse.

- #categories – identifies the topic hub
- #emotion #emotion_name, number, number
 - This is used to tell the animator in Unity that the virtual patient needs to play a specific facial animation blendshape. The first number tells the system how long the animation should play over the next dialog chunk (as a percentage of that dialog's duration), and the final number is the intensity of the emotion.
 - If the code is #emotion #smile, 100, 70, Simone will smile for 100% of her next line of dialog with an intensity of 70%.
 - Generally, anything above 70% is a bit much, but feel free to play around with the values.
- #gesture #AnimationName
 - Informs Unity animator to play a specific Simone animation at this point of the dialog.

(See **NXHUMAN Art Asset List** for full list of possible animations and facial blendshapes)

- #developComSkills – used for first learning objective in summary screen, pairs with VAR developComSkills.
- #seqTaken/#bseqTaken – used for the second learning objective, identifies the user's sequence in the encounter and pairs with VAR seqTaken.
- #explDiseaseStates, #dietAndExer, #coveredTopicName, etc. – provides feedback on how much the user has explored a given topic, and updates the final score accordingly.
- #finalWords – delineates the final feedback section of the summary screen.

Functions

- updateVal
 - Allows us to quickly add one variable's value onto another. Mainly used in the AAR.
- calcDate
 - Using the variables age, currYear, and bYear, allows us to set the virtual patient's birthday relative to current date.
- Tunnel_as_thread
 - Allows for threading content from one topic to another, making it so the new content is loaded to the user from whichever topic they last visited. Mostly used in encounter 5 and the preceptor feedback.
- Patient_responses
 - This is *technically* a sub-section within introductions but is essentially a giant function.
 - This allows the user to speak as many or as little of the six introductory dialog options as they want, and the function will combine the possible responses for the virtual patient to reply with.

Virtual Patient Encounter Script Structure

- For example:
 - STUDENT: “Hi Simone. I’m a student pharmacist. Is it okay if we talk?”
 - PATIENT: “Nice to meet you. Yes, it’s okay if we talk.”
- In the above example, the student’s dialog are three separate options in the ink script, meaning that the patient would only reply one at a time in Inky.
- The student requested permission to counsel, said hello, and introduced themselves. The function then has the virtual patient reply with those three temp variables.
- Temp variables (~ temp_saidHello, etc.)
 - Specifies the dialog content for a given response.
- After parsing the responses, it next informs the topic hub variables that potentially multiple options have been selected, and shows or hides the appropriate dialog option(s).
- Increases trust value based on the student’s dialog & temp variables.
- Finally, returns the user to their previous location in the dialog tree (in this case, the topic hub).

Variables

Setup

- Age, currYear, bYear – virtual patient’s age, the current year, and the blank variable for the patient’s birthyear.
- Trust & maxTrust
 - How much the virtual patient trusts the student/user. This is altered by the user’s choices/dialog throughout an encounter. Also feeds into the AAR.

Speakers

- Purpose of speakers is to inform scripts in Unity which character (virtual patient, student) is using the current dialog. That’s it.
- Three speakers: PATIENT (VAR P = “PATIENT: “), STUDENT (VAR S = “STUDENT: “), and SUMMARY (VAR Su = “SUMMARY: “).
 - Summary ties into the after-action review (AAR) screen that’s shown at the end of every encounter, and is how we fill in the correct feedback text.

Introductions

- First six variables are for activating the correct dialog responses in patient_responses
- Following five are to close/hide introduction options after selection.

Topic Variables

With some exceptions, every topic contains the following:

Virtual Patient Encounter Script Structure

- At least one counter
 - Generally used to determine when the user has selected all possible dialog options within a given topic, setting another variable to true (e.g., coveredSocialHistory), and sending the user back to the topic hub. The option to return to that topic is now hidden from the user.
 - Occasionally used to alter patient responses within certain topics, such as BRCA tests, or adding a new medication.
- Covered [name of topic]
 - Used to hide a topic option in the topic hub once all possible dialog options for a given topic have been exhausted.
- Contextual variables
 - Used to show/hide content.
 - For example, the option to discuss the patient's smoking habit is not shown until the user specifically asks the patient if they smoke.
 - As many or as little as needed per topic.

Recap/Encounter Close

- developComSkills – set to true if the patient's trust level is greater than a specific number.
- Score & maxScore – used to determine the user's score for a given encounter.

Sequencing

Gathers a count on how many topics the user has selected and adds content from LIST SeqOptions the empty variable seqTaken. Originally the plan was to use this setup to have an exact list of the user's choices during an encounter, but this is not possible due to limitations in Ink and Inky. The summary section makes a check to see that seqTaken at least has the options, or counts the size of seqTaken's contents. Discussed more in that section.

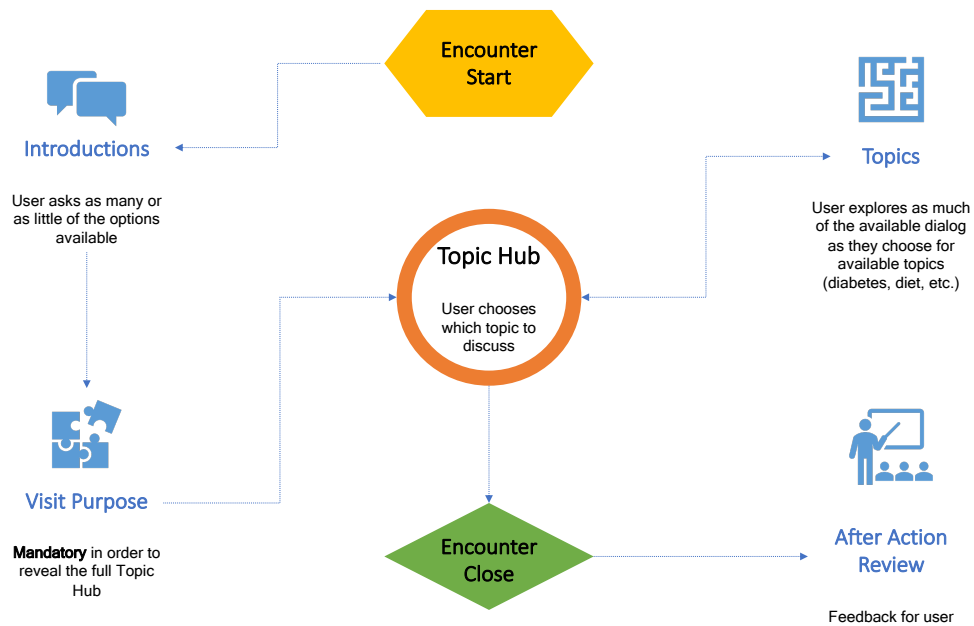
- LIST SeqOptions – contains the following options that are then added to seqTaken: greeting, confirmID, meds, brca (enc 4 only), meds, medCounsel (only if multiple medications are prescribed), socialHistory, purpose, diseaseStateOne-Three, pharmacoTherapy (not in enc 1), nutritionalTherapy, physicalActivity.
- VAR seqTaken – an empty variable that has content added when a dialog option is selected.

Structure

User Flow

The following visualizes a general topic branch the user takes in nXhuman. This does not include specific dialog options as this is not viable due to the breadth of options available.

Virtual Patient Encounter Script Structure



Encounter Start

Contains a brief verbalization for the patient and a redirect to the topic hub. This section is meant to be short and to the point.

Virtual Patient Encounter Script Structure

Introduction



Figure 1 Introductory options in-game

This is where the user can introduce themselves to the patient, say hello, ask the patient how they're feeling, request the patient's permission to counsel, ask the patient for their identifiers (full name and birthdate), and directly ask for the purpose of the visit. Except for the final option, every choice raises the patient's trust value.

Once the user has asked for the purpose of the visit, the introductory options are no longer available. The user is now in the meat of the discussion and the After-Action Review will address whether they properly introduced themselves to the patient.

In Inky, the user can only select one introductory option at a time. Within a nXhuman build users can speak multiple introductory options at once (e.g., "Hi Simone, I'm a student pharmacist, is it okay with you if we talk?") or select one option at a time within the UI.

Finally, the ability to end the encounter is only available once the user has asked for the visit purpose.

Visit Purpose

Effectively a key for the rest of the encounter. This is where the patient explains why they have visited the clinic today – their primary complaint – and ranges from fears of developing

Virtual Patient Encounter Script Structure

diabetes to consultation on medical tests. After the user has asked for the visit purpose the entire topic hub is unlocked.

Topic Hub

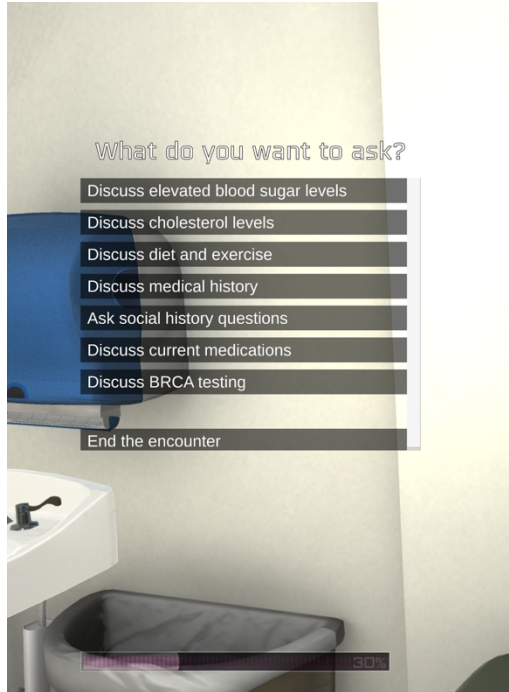


Figure 2 Topic hub in-game

As the name suggests, this is where users select a specific topic to discuss with the virtual patient. nXhuman is structured so the user needs to inquire into all aspects of the patient's life in order to have enough information to make a determination on the patient's therapy. As such, every topic of discussion is given its own branch in the dialog tree and the topic hub.

Apart from the introductory choices ("Say hello" through "Ask about visit purpose"), every topic's visibility to the user is conditional (see the ink code below).

```
516 -> DONE
517
518 /*-----
519 | | TOPIC HUB
520 | |-----*/
521 == topic_hub ==
522 //Leave #categories
523 #categories
524
525 * {TURNS_SINCE(->encounter_start) < 1}[Say hello] -> introductions.greet
526 * {not coveredPurpose && not visitedFeelings}[Ask how patient is doing] -> introductions.feeling
527 * {not coveredPurpose && not visitedIntroduce}[Introduce yourself] -> introductions.introduce
528 * {not coveredPurpose && not visitedPermission}[Ask permission to counsel] -> introductions.permission
529 * {not coveredPurpose}[Confirm patient identity] -> introductions.ID
530 + {not coveredPurpose}[Ask about visit purpose] -> introductions.visitPurpose
531 /* {TURNS_SINCE(->visit_purpose)=0}[Express support] -> brca_testing.reassure
532 + {coveredPurpose && not coveredBRCA}[Discuss BRCA testing] -> brca_testing
533 + {socialVar1 && not coveredSocialHistory}[Ask social history questions] -> social_history
534 + {lifeVar1 && not coveredDietExer}[Discuss diet and exercise] -> diet_and_exercise
535 + {medHistVar1 && not coveredMedHist}[Discuss medical history] -> existing_diseases
536 + {currMedsVar1 && not coveredCurrMeds}[Discuss current medications] -> currMedications
537 + {secondDiseaseVar1 && not coveredSecondDisease}[Discuss cholesterol levels] -> high_cholesterol
538 + {newMedsVar1 && not coveredNewMeds}[Discuss elevated blood sugar levels] -> newMedications
539 + [***] -> topic_hub //spacer for Unity
540 * {coveredPurpose}[End the encounter] -> encounter_close
541
```

Figure 3 Topic hub in Inky

Virtual Patient Encounter Script Structure

In addition to the introductory options, the topic hub contains the following:

Social History

Contains general background questions that are followed up in appropriate topics. Users can ask the patient questions about their alcohol intake, smoking habits, occupation, caffeine consumption, illicit drug use, and more. Functionally the verbal equivalent of the questionnaires we fill out in a doctor's office.

Medical History

Here the user asks about the patient's past medical history (PMH), including family medical history (FMH), and any treatment for disease states. In nXhuman, this ranges from mainly focusing on the FMH to the patient having multiple disease states.

Medications

Explores the patient's medications, current and newly prescribed, dosage, adherence, possible side effects, and more. This includes over the counter medications such as painkillers, and vitamins or herbal supplements. The goal of this topic is to evaluate the patient's medical therapy and determine if the therapy needs to be altered.

Disease States

Generic term for any condition that is afflicting the patient. Ranges from diabetes to high cholesterol levels. Each disease state is more or less functionally identical in structure, but has a dedicated topic for each specific condition.

Let's look at Simone from nXhuman as an example. In her fourth encounter she's afflicted with:

- Type 2 diabetes, with uncontrolled blood sugar levels
- High blood pressure, currently controlled with two medications
- Recently diagnosed with high cholesterol

Each disease state has its own option in the topic hub, and it's dependent on the user to uncover lifestyle changes in addition to therapy changes for the patient.

Diet & Exercise

(Note: this is split into two sections in encounter 1 due to the greater focus on the patient's lifestyle. Every other encounter combines the two.)

Virtual Patient Encounter Script Structure

Here users uncover details about the patient's diet and exercise habits (if any), and where changes can be made in order to benefit the patient's health. For example, in encounter 4, Simone has stuck to a low-salt diet for her blood pressure, but consumes a high quantity of fatty meats and dairy products. Discovering this will allow users to provide healthier dietary options to lower Simone's cholesterol levels.

Per our work with subject matter experts (SMEs), nXhuman emphasizes lifestyle changes in addition to any possible therapy interventions. In some encounters there is a greater focus on lifestyle changes, so this topic is generally one of the larger subjects in any given encounter.

Spacer

Wondering what the [***] option is doing? It's merely a spacer/placeholder to spread out the dialog options from an option to bring up another topic or end the encounter. Our code in Unity looks for the asterisks, and then hides it. In Ink it functionally sends the user back to the topic hub, should they feel compelled to click on it.

Encounter Close

Once users have discovered the patient's primary concern, they can end the encounter at any point. This could have a disastrous impact on their score, depending on how early they end the encounter, but the option is at their disposal.

This section contains the following options:

- Recap the encounter to the patient. Shows or hides student dialog depending on the options selected throughout the encounter. Raises the patient's trust value.
- Ask for any final questions. Offers the patient a chance to address any lingering questions, or clarify anything covered during the encounter. Raises trust value.
- Say goodbye. Ends the encounter and whisks the user to the AAR. The student's dialog is altered based on their trust level. High trust leads to a warm farewell, while having no trust at all has the student leaving the room without saying a word.

Virtual Patient Encounter Script Structure

Summary & Feedback

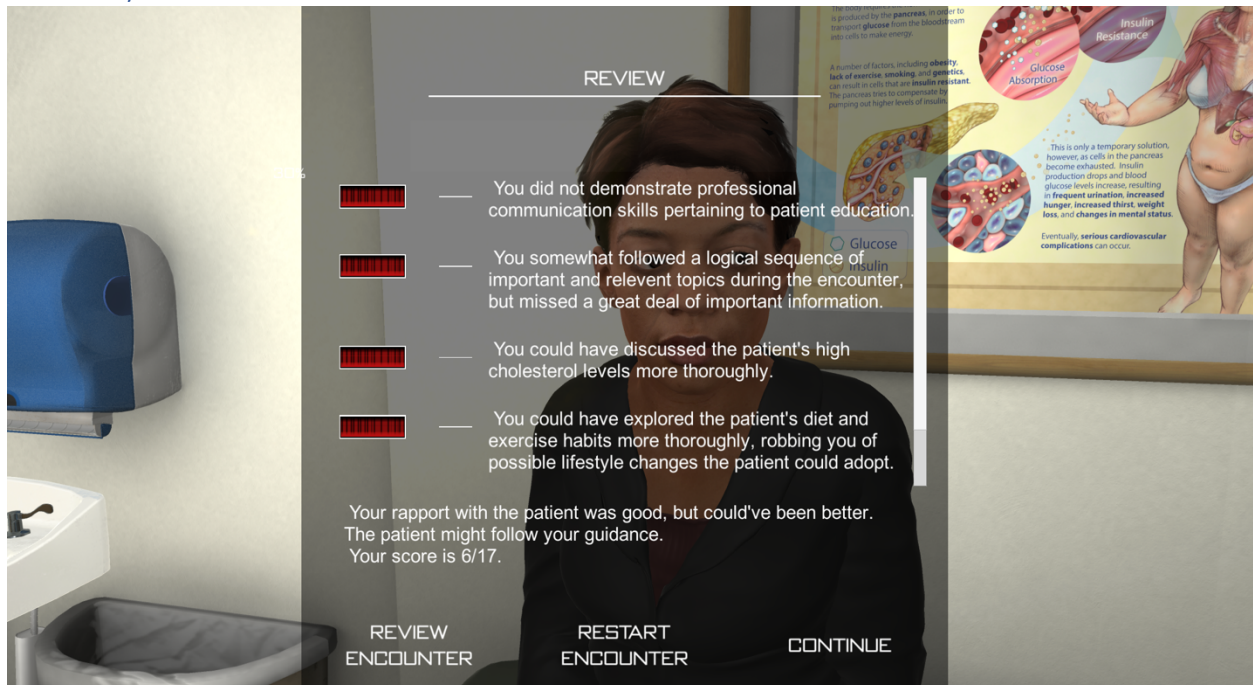


Figure 4 After-action review in-game

This section gives feedback based on the user's trust score, the variable seqTaken, and either specific topic counters (e.g., diabetesCounter) or if certain variables are true (such as coveredDietExer). The review screen (figure 3) then goes through several learning objectives and provides feedback to the user. Finally, the user is awarded a score for the encounter.

- Trust level is first used for the first learning objective – demonstrate professional communication skills. If it is above a certain level this is achieved and vice versa. Increases the users score if accomplished.
- Following a logical sequence of topics is based on seqTaken, and whether it contains specific content, or if seqTaken is greater than or equal to the number of options possible. See figure 4 below for the Ink code.
 - This updates a variable estProperSeq, and is eventually fed into the final score.
- Checks if certain topic variables are true, meaning the user has exhausted all options in that given topic, or if a specific topic's counter is higher than a value. Increases the score or not depending on the results.
- Finally, provides qualitative feedback based on the user's trust value and presents the user's final score.

Virtual Patient Encounter Script Structure

```
1127 =summary
1128 {trust >= 6:
1129   ~ developComSkills = true
1130   ~ score++
1131   {Su}You demonstrated professional communication skills pertaining to effective patient
        education. Good work! #developComSkills
1132 -else: {Su}You did not demonstrate professional communication skills pertaining to patient
        education.#developComSkills
1133 }
1134
1135 {LIST_COUNT(seqTaken) >= 9:
1136   ~updateVal(estProperSeq, 4)
1137   {Su}You established a logical sequence of important and relevant topics during the
        encounter and addressed the patient's concerns.#seqTaken
1138 }
1139 {LIST_COUNT(seqTaken) < 9 && LIST_COUNT(seqTaken) >= 6:
1140   ~updateVal(estProperSeq, 3)
1141   {Su}You mostly followed a logical sequence of important and relevant topics during the
        encounter, but did not address all the patient's concerns.#seqTaken
1142 }
1143
1144 {LIST_COUNT(seqTaken) < 6 && LIST_COUNT(seqTaken) >= 3:
1145   ~updateVal(estProperSeq, 2)
1146   {Su}You somewhat followed a logical sequence of important and relevant topics during
        the encounter, but missed a great deal of important information.#seqTaken
1147 }
1148
1149 {LIST_COUNT(seqTaken) < 3 && LIST_COUNT(seqTaken) > 0:
1150   ~updateVal(estProperSeq, 1)
1151   {Su}You barely covered any relevant topics during the encounter.#seqTaken
1152 }
1153
1154 {LIST_COUNT(seqTaken)==0:
1155   ~estProperSeq = 0
1156   {Su}You did not even confirm the patient's identity or request permission to counsel.
        This is unprofessional and may place your institution in legal jeopardy.#seqTaken
1157 }
1158
1159 {coveredSecondDisease:
1160   ~score++
1161   {Su}You helped steer the patient towards lifestyle changes that can lower their
        cholesterol levels.#coveredSecondDisease
1162   -else:{Su}You could have discussed the patient's high cholesterol levels more
        thoroughly.#coveredSecondDisease
1163 }
1164
1165 {coveredDietExer:
1166   ~score++
1167   {Su}You thoroughly investigated the patient's diet and exercise habits.#coveredDietExer
1168   -else:{Su}You could have explored the patient's diet and exercise habits more
        thoroughly, robbing you of possible lifestyle changes the patient could adopt
        .#coveredDietExer
1169 }
1170
1171 {coveredNewMeds:
1172   ~score++
1173   {Su}You correctly explained the patient's new insulin dose. #coveredNewMeds
1174   -else:{Su}You should have given more guidance on the patient's new insulin dose.
        #coveredNewMeds
1175 }
1176
1177 {trust >= 6: {Su}Your rapport with the patient was excellent. You were empathetic at the
        appropriate times, listened to the patient, and provided guidance. The patient will
        likely follow your advice.}#finalWords
1178 {trust < 6 && trust >= 3: {Su}Your rapport with the patient was good, but could've been
        better. The patient might follow your guidance.}#finalWords
1179 {trust < 3 && trust > 0: {Su}Your rapport with the patient needs work. The patient will
        likely not adhere to your guidance.}#finalWords
1180
1181 {trust == 0: {Su}You did nothing to gain the patient's trust.}#finalWords
1182
1183 ~updateVal(score, estProperSeq)
1184 ~score = score + trust
1185 {Su}Your score is {score}/{maxScore}.
1186 {score >= maxScore:
1187   {Su}Awesome work!
1188 }
1189 ->DONE
```

Virtual Patient Encounter Script Structure

Figure 5 Summary screen/after-action review in Inky

Hashtags and Variables

The hashtags and variables are *vital* to functioning summary screen, at least in Unity. Our code looks for the hashtags, the variables *with the same name* as the hashtags, and then aligns the dialog in its proper place.

(See **Technical Content, Hashtags** for a list of hashtags in use and our naming schema)

Use the following in order to have a nice, working summary screen.

1. Remember speakers? This is where Summary comes into play. Every bit of text feedback must first be marked with {Su}.
2. Hashtags must be marked after every line of dialog. This way the system knows what text to look for.
3. True/false variables are needed for the summary screen.
 - The user either accomplished a learning objective, or did not meet its requirements.
4. **Variables and hashtags must have the same name.**
 - This is bolded for a reason. As an example, the first learning objective looks for #developComSkills, which happens to be the same name as the variable developComSkills. Yes, this is spelling and case sensitive; I've made that mistake enough times for one project.
5. #bSeqTaken/#seqTaken are used for anything involving the user's sequence in the encounter.
6. Want to see if the user fully explored a given topic? Use the same variable that hides an option in the topic hub.
 - For example, coveredNewMeds is set to true when the user has asked every question about the patient's new medication.
 - This follows the naming convention of coveredTopicName.
 - Just make sure to include a hashtag with a matching name 😊.
7. #finalWords **can only be in the final section** of the summary.
 - This is so the text, like the name implies, shows on the bottom portion of the summary screen.
 - If you **don't** follow this, it can *break the entire screen layout*.

Following all this, the bottom portion of Ink code updates the user's score with their trust value and estProperSeq, and assigns the user a final score.