# IT 140 Transcript for Sample Dragon Text Game Walkthrough

[00:00:06.03] Hello. In this video, you will see a sample text-based game, similar to the one that you will design and code for Projects One and Two. This sample game has a dragon theme, but you will choose your own theme for the game that you design. This video will help you understand how a text-based game works, and you'll get a chance to see the sample dragon text game in action.

[00:00:27.47] The text game involves a player entering commands to move to different rooms, or get different items. The goal of the game is for the player to get all of the items before they encounter the villain. For Project One, you'll be designing the text-based game. As a part of your design, you will be asked to create a storyboard, which includes a description of the theme, as well as a map. You will also be asked to create pseudocode or a flowchart to help you with the two major processes in the game—the process that allows the player to move between rooms and that allows the player to get items from different rooms.

[00:01:05.43] In Project Two, you will move beyond your designs and actually develop the code for the game. But before we get into that, let's first look at the sample storyboard for the game, which you can also find in the Project One Supporting Materials section.

[00:01:22.49] As you can see, there's a short paragraph that describes the theme of the game—in this case, the dragon theme. But again, you will be creating something on your own designs. And please choose something that you're interested in. Perhaps it could be space. It could be a spy theme. You could choose anything that has the idea of rooms, items, and a villain, so make sure to choose something that interests you. In this case, all of the different items and the rooms sort of fit with the dragon theme. It's sort of like you're in a castle, trying to navigate different items that you might need to defeat the villain.

[00:01:51.77] Once you've described the theme and the different items and rooms in your game, you want to lay them out in this map. And you've been given a blank map in either the Design Document or the Design Presentation Template, depending on which way you want to present your designs. And you can fill it in with all of the different rooms and items that go with your theme.

[00:02:11.85] So you can see here that all of the different rooms are laid out. And you can clearly see the item that's in each room. The great hall, or the start room, does not have an item in it. And the only other room that doesn't have an item is the dining room, which contains the villain—in this case, the dragon. You can see that this map also lays out the different directions, so that you can easily see how the rooms are connected to one another. This is going to be really helpful when you actually get to Project Two and start both developing and testing your code.

[00:02:43.07] Speaking of the code, let's open up PyCharm where I have some sample code. You'll notice that there is some actual code in here. Some of this has actually been given to you in [the] Project Two directions, as well, for you to use as a sample and a reference. But there's a lot of spaces where the code has been removed and just commented out. That was just for the purposes of this video.

[00:03:02.33] So we start with a couple of different functions. You can organize it in a different way, perhaps as part of one function. In this game, I've used two functions. The basic functionality, though, is that I want to tell the player the basic rules of the game. So in this case, it's that the player knows that they have to collect six items to win the game, or they will be eaten by the dragon. I have to be able to tell them the different commands that they can make, such as “go south,” “go north,” “go east,” “go west,” and “get” whatever the item is in the room that they're in, so that they can add it to the inventory.

[00:03:37.48] I also have this separate function here which, again, most of it is commented out. And it would show the current room of the player, their inventory, and the item that's in the room that they're currently in, if one exists.

[00:03:50.16] Next, the main gameplay actually takes place in the “main()” function. So you can see here, I have my “main()” function defined. And then I would have a section of code where I would define the inventory, which starts off as empty, because the player has not collected any items.

[00:04:05.92] Then I have a dictionary. And a sample version of this has also been included in the Project Two guidelines as well. Again, you will have to customize this, because it needs to fit the theme of your game. So you can see here that using your map that you did as a part of Project One will be really helpful when you develop the code for Project Two.

[00:04:23.08] You can see how all the rooms are linked together. So for example, from the great hall, if I were to go south, I would hit the bedroom. So you can see south, bedroom. You also want to make sure that you link the different items to the rooms. So for example, for the bedroom, the item in the bedroom is armor. And you can see that I've listed that as my last link in this dictionary. So it's important to make sure that you use your map to set this up properly.

[00:04:47.74] Next in my game, as part of the game play, I would start the player in the great hall, which is my start room. In your design, it might be different. I would have a call to show the player the game instructions. And then I have this while loop that will continue to handle the different commands from the player— so continue to prompt them for a command and then actually do whatever process is needed, based on whatever command that they enter.

[00:05:11.89] It's important that I have conditionals that help me exit the loop, once the player has either lost the game (for example, by encountering the dragon) or won the game by collecting all of the different items for the game. So make sure to include exit conditions for your while loop as well. You can see that I also have some decision branching that lays out what I should be doing if players type the “go” command or type the “get” command. And that's something that you'll be working on as well.

[00:05:40.25] So let me show you for a minute just how this code will actually run, so that you can see a little bit better how the game works. So I'm going to go to the actual solution, right-click, and click “Run.” And this is something that you definitely want to do when you're creating your own code, to make sure that it's working properly and that you've tested it.

[00:05:56.98] So you can see, as expected, when I start up the game, It prompts me. It tells me the directions of the game. It tells me the different commands that I can enter. It tells me where I am and the inventory. Then it prompts me to enter a move.

[00:06:09.10] So if I'm in the great hall—let's say I want to get the book—I can use my map to help me navigate and test the game. Again, remember, if somebody was actually playing this for real, they wouldn't have access to the map. That makes it a little bit more challenging for them. But since I'm developing the code and testing it, I want to actually have the map up, side by side, just to make it a little bit easier and to make my testing more efficient.

[00:06:30.52] So I'm going to try to type “go west.” And when I go west, I can see that I'm in the library. That means my game is working. It's moved me to the correct room. So next, I would get a book—the book, which the item in that room. And you can see that the book has now been added to my inventory.

[00:06:47.11] What happens if I type something that's not in there, like the shield? It should give me an error message. And you can see that it does. It tells me that I can't get the shield. So as you're developing your game, make sure that you use input validation, by including error messages and things like that for when the player types an improper command.

[00:07:06.88] So now, as I'm testing, normally I would test a bunch of different rooms, commands to go back and forth, collecting different items, just to make sure that it all works. I would also definitely want to test what happens if I tried to win the game and what would happen if I actually lost the game by not getting all of the items before I met the dragon. So for the sake of time, I'm just going to show you that option of what happens if I encounter the villain first.

[00:07:31.70] And you can see here that I'm in the library. So I want to get to the dining room. So using my handy dandy map, I can type—instead of “get east,” I want to type “go east.” And now, I'm in the great hall. I want to go east again. I'm in the kitchen. May as well get the sandwich while I'm in there. And then if I go north, I should encounter the dragon, which I did.

[00:07:54.91] You can see that the game tells me that I saw a dragon. “Nom, nom, game over.” And it's actually exited the game, and the game is now over. And it gives me a little message to let me know that.

[00:08:04.90] So that's the basics of how a text-based game works. So I hope that helps you understand it a little bit better and gives you an idea of what you will be designing as part of Project One and also developing code for as a part of Project Two.