CS 240: Terminal Control Codes Transcript

[00:00:00] So how do we create a console user interface? Well, really all we have to do is print special strings to the standard output or system dot out.

Start visual description. The professor demonstrates how to create a console user interface by printing special strings to the standard output. The professor explains that certain strings have particular meanings to the console, such as changing the background color or clearing the screen. End visual description.

[00:00:10] And if we print certain strings, they have particular meanings to the to the console.

[00:00:15] For example, if you, if you print a particular string to the output, you can tell it to make the background color yellow or you can tell it to clear the screen.

[00:00:26] And so there's these things called control codes.

[00:00:29] Now when I talk about a control code, it's really just a special string.

[00:00:32] So just think of strings that have certain special characters in them that have special meaning to the to the console when you print them.

[00:00:39] And so you can do things with these uh control codes like clear the screen or clear the terminal and you can set the cursor position. So, there is a notion of a cursor position which is where the output appears. Uh The next time you print from your program, wherever the cursor is, is where that output is going to appear.

[00:00:58] So you can, you can move the cursor around.

[00:01:01] Um you can set the background, color, the text color.

[00:01:04] Um You can also set the uh attributes of the text like bold italic underline, et cetera.

[00:01:09] And if you really want to know the gory details of how this works, you can go to Wikipedia and there’s a, there’s a web page there that that really goes through um all the details far more than you probably want to know about it.

[00:01:22] But if you’re interested, you can learn more about how these control codes work.

[00:01:26] Let’s take a look at how these control codes work here. Let’s uh here’s a, here’s a file that we actually give you uh for the chess project. It’s kind of small, but all this, this escape sequences class, that’s another term people sometimes use for control codes, they call them escape sequences.

Start visual description. The professor shows a file for the chess project that contains an escape sequences class. This class defines constants that control the console's behavior, such as the Unicode escape character and the erase screen command. End visual description.

[00:01:46] And so you can see here that in this escape sequences class, we just define a bunch of constants that have special meaning to the, to the console. So, um the very first constant is called Unicode escape. So anytime we print a string to the console that’s intended to control the behavior of the console, um the first character we, we print is a Unicode escape character.

[00:02:10] So slash U 001 B and that kind of tells the console that whatever characters come next are going to tell, tell it what to do.

[00:02:19] And then um you can see down here there’s a constant called erase screen.

[00:02:24] So erase screen is just printing the Unicode escape and then printing a left square bracket and a capital H and then printing another Unicode escape and then printing a square bracket, left square bracket, uh A two and then a capital J.

[00:02:37] And so you can kind of see uh what those escape sequences or those terminal codes look like.

[00:02:43] Now, they're not intended to really make any special sense.

[00:02:45] I mean, if you read that Wikipedia article, you can really see um what these strings mean.

[00:02:50] But, but for the most part, um you just want to use these constants that we provide in this class. And then you can see uh down below what, what uh control codes do you need to write to, you know, make the text bold or faint or italic and you can see how to set the background, color, the text color and so forth.

[00:03:10] And so really just with the system dot out dot print line, you can, you can control everything you want to control about your terminal output.

Start visual description. The professor demonstrates how to use the system.out.println method to control terminal output, including setting text attributes like bold, italic, and underline, as well as changing background and text colors. End visual description.

[00:03:17] And of course, this is all about output the input for this kind of program, of course comes from the keyboard.

[00:03:24] So how do you use those low-level terminal codes and actually draw something on the screen? We have a code example that we provide uh to show you how to do this.

[00:03:36] So what we, what we have here is a program that prints a tic tac toe board in the console.

[00:03:41] Now, this has obvious similarity to a chess board but, but it's quite different as well.

[00:03:46] And so let's just take a look at the source code for that, that program that prints the tic tac toe board.

[00:03:52] So we'll go to intellect here.

[00:03:54] Of course, this code sample is in the um GitHub repository for CS 240.

[00:04:00] So the, the class here is called Tic tac toe.

Start visual description. The professor runs a program that prints a tic-tac-toe board in the console. The program draws the board, sets colors for the headers and squares, and uses control codes to change the appearance of the output. End visual description.

[00:04:04] And all this program does is it, it uh it draws the board draws a tic tac toe board and it draws random Xs and Os in the different cells of, of the board.

[00:04:17] And so if you go to the main function here, you can see that um all this program does is first of all, it draws the headers.

[00:04:24] Now, what are those headers? Let's actually run it so we can see what we're talking about.

[00:04:28] So if I run this program, it prints, it prints this.

[00:04:33] So across the top here, you can see uh in green text on the black background, it says tick tac toe. So those are the headers.

[00:04:40] And then of course, we have the squares that make up the, the tic tac toe board.

[00:04:43] And so if you look at the code, first thing it does is it draws the headers tic tac toe and then it draws the board.

[00:04:53] And um now, before we dive into that code, let's look a little bit farther down to see some of the functions that we have um down, down at the bottom, we have a function called set white, for example.

[00:05:09] So white is the background color for the actual squares on the board.

[00:05:14] And so in, in that case, I want to draw essentially white spaces to, to make a, a white square.

[00:05:22] And so the way I draw a white space is I, I send these control codes to the, to the console.

[00:05:29] Um The first control code sets the background color to white.

[00:05:33] And then the next control code sets the text color to white.

[00:05:36] And then after that, everything that I print is going to be white on white.

[00:05:41] So you won't actually be able to see it.

[00:05:43] Um So I'm just going to print a bunch of spaces that are white on white and that will produce the background for those, those squares on the board.

[00:05:51] Um The red um If I want to draw red, for example, you can see the dividers between the squares on the board, they're red.

[00:05:58] So if I want to draw those horizontal and vertical lines, I need to set the, the background and the text colors to red.

[00:06:06] And so I just call this little function to do that.

[00:06:08] Um And then I also have a function to set the colors to black. You'll notice here that um actually, most of what we printed out here is actually black.

[00:06:19] And so um to print that I, I set both colors to be black. So, it would print out that way.

[00:06:25] And of course, to print out the headers up here, I had to set the background to black and then the text color to green.

[00:06:30] And so you kind of get the idea.

[00:06:32] So to change the color of what I'm about to print, all I've got to do is send a couple of, of codes to the terminal and then it knows what to do from there.

[00:06:40] So let's, let's take a little bit uh closer look here.

[00:06:44] So the first function we called was print, header, print or draw headers.

[00:06:53] And so the first thing I do is I set, set it to black.

[00:06:59] And then what I do is I um have a, an array here that has the header tech and then I just go through a loop and I loop through all the headers and I, I draw each one and um between the headers, I also have to print a spacer, the spaces that, that uh are in this spot right here.

[00:07:23] Um I need to print those as well. So really, it's just a matter of making it black, drawing some spaces that are black on black and then changing the text color to green, then I draw tick and then I set everything back to black and draw some spaces and then so forth.

[00:07:39] So you just got to make sure that you have the right settings every time you want to print something.

[00:07:42] So it shows up in the right colors.

[00:07:45] And uh if we go to the draw header function, it just kind of figures out uh how many spaces do I need to print to the left of the header text? And then it prints out the header text and then it prints out some spaces after the header text.

[00:08:00] And when we print the header text, you can see that it, it sets the background color to black and the text color to green.

[00:08:08] So it's a pretty simple idea.

[00:08:12] Now, drawing the board is a little bit more uh challenge, challenging than drawing the headers.

[00:08:17] But um the code is not so bad.

[00:08:20] So here's the draw tic tac toe board method.

[00:08:24] And in this case, I'm going to draw um well, we have a three-by-three board. So, I need to go into a loop that draws three rows.

[00:08:36] And so for each time around the loop, we call draw row of squares to draw that row.

[00:08:42] And then until we get to the bottom row, we want to actually also print out the, the separator, that red separator that, that separates the rows.

[00:08:51] And so if, if we're not at the very bottom of the board, we're going to go ahead and draw that red, that red line that separates the two rows.

[00:08:59] And so that's what uh the draw horizontal line method is.

[00:09:03] And uh you could read the, the details of those methods to see how that goes.

[00:09:07] But uh it's pretty much what you would expect.

[00:09:11] So, in this case, if we want to draw a horizontal line, um the thickness of these dividers is, is a constant as well.

[00:09:19] And for each, we're going to draw several rows of spaces to draw that divider.

[00:09:25] And, and so you see here that we just go through a loop and for each time around the loop, we set the color to red, then we print a bunch of spaces and then we set it to black and then we do a print line.

[00:09:38] Now, this is interesting right here because it, it's kind of um curious why would I set the color to black? But you'll notice here that when we're printing the, the spaces in the divider that we want those to be red, but we want everything else after that to be black.

[00:09:54] And so what we have to do there is set the, the, the background and the text colors to black.

[00:10:00] And then when we hit, when we print the print line out, the rest of the line is, is painted with black.

[00:10:08] So that's one thing you need to realize is that when you print a new line character, whatever the current settings are, is what will be used to fill out the current line that you're already on.

[00:10:17] And so you need to, to watch that as well.

[00:10:20] And so, um yeah, that prints the, the horizontal line.

[00:10:25] Of course, up here, we have some logic that draws a rule of squares and you can read through that and study it.

[00:10:31] Um But, but if you'll study this example and understand it, that'll help you um know how to draw your chessboard for your, for your project.

[00:10:39] OK? I just talked about um this helpful hint here about printing new lines, so I don't need to repeat that.

[00:10:45] So I think that's uh pretty much all you need to know about control codes or terminal codes for your uh chess project.