

Instructional Video Project Procedure

This video project was a very complex and time consuming task to complete.

The first thing I did was purchase a LEGO Mindstorms NXT robotics kit from the online LEGO store. I had been wanting to purchase one for a while, and this video project more or less convinced me to go through with it.

The next thing I did was create a storyboard and rough draft of a script in Microsoft PowerPoint. I typed the storyboard items into the slide formatted as bullets, and I typed the corresponding script for each slide in the notes area. This took a considerable amount of time because I had to walk through every video action in my head as I typed it out. Many slides were constantly modified and reworded as time went on.

Once I was happy with my storyboard and script, I printed them out and borrowed a camera and tripod from the college. I used a room in my house as the film location, which required the use of a fold out table, a chair, and the LEGO NXT kit.

I had to film everything myself, which required me to press the record button and then walk onto the scene and recite my lines from memory or (in scenes where my face wasn't showing) from the script printout. I tried to keep the camera at just above my waist level so it wasn't looking down at me when I was present in the scene sitting at the table. For most of the other angles, I had it sitting next to the table and looking down at the pieces. I was able to use the camera's flip-out screen to easily fine-tune its position while sitting where I needed to be.

The opening title scene of the video shows a robot car that stops before hitting some old-style LEGO pieces. This is a robot I had already built and was in the process of programming. For the purposes of the video, I programmed it to stop driving forward when it detected something in front of it. I wanted it to be more elaborate and sophisticated, such as to steer around the old-style pieces, but that would have taken much more time.

In the first video, to film the NXT output readings when demonstrating the sensors, I used my smaller digital camera's video recording feature. This required me to have two cameras recording at once, knowing I would have to edit their recording to be in sync later on.

For the editing process, I used Ulead's VideoStudio 9 SE DVD software. This software automatically downloaded the film video and sound from the DV camera into a RAW AVI file over 6 GB in size. For the title screen, I used a video filter to create the colored pencil effect, and I used title text to show the title. I also used title text all throughout both videos to label items on the screen that I felt were important, such as sensors or pieces.

To create the NXT sensor demonstration footage, I used my digital camera's recordings as a video overlay onto the main DV camera's footage. I used the sound playback from the recordings to synchronize the two videos together in time. I then muted the digital camera's video so only the higher quality DV camera's audio would be heard. For most of the audio all throughout the video, I had to increase the volume in the editing software to make it more audible.

For the robot building part of the first video, I actually filmed myself completely building the robot, which took about 15 or 20 minutes. I then sped this footage up to be 10 times faster with the editing software (it's maximum speed increase), but this didn't prove to be as fast as I wanted it because it still resulted in over a minute's worth of video. I then saved this portion of the video fully sped up as a new video file, and increased that new file's speed by about 7 times or so. This achieved the speed and time I was wanting while still retaining the effect I was going for. It took me a long time to find a song that sounded good for this part of the video. However, I eventually found a portion of a song that had just the right amount of instrumental music and cutoff before it began with vocals.

Speaking of music, I originally wasn't going to have any in the video except in the title screen and in the robot building scene. However, I found the video to be somewhat boring and lifeless without some form of background music, so I searched and found a few songs that each had a portion of music that didn't include lyrics, had a nice and calm beat, and could be looped for a long time without becoming annoying to the listener. I used the video editing software to tweak the beginning and ending time for one of these loops and then copied and pasted the loop over and over to achieve the seemingly endless looping music. I edited each loop to have a very low volume value so it would stay in the background and allow my voice to be the main focus throughout the video. Only in the introductory title, the robot building scene, and in the ending credits did the music play at a normal volume.

In the second video, where I demonstrated on-screen programming with the NXT software, I used a free screen recording software package I found online. It is called CamStudio, and can be downloaded from:

http://sourceforge.net/project/showfiles.php?group_id=131922&package_id=144908

I set it to record a region of the screen which contained the software. In order to remember my lines while still moving the mouse cursor to the necessary positions, the recording process required me to make about 8 different recordings that I later spliced together with the editing software. I used my computer microphone to record my voice during these recordings, which was an included feature in the CamStudio software.

Also in the second video, I featured a scene where the robot is tested and the program action that is currently being executed is shown as an overlay on the table. To accomplish this, I took screenshots of each of the programming blocks from the programming software and edited them in Paint Shop Pro X. This really only consisted of cropping and saving each as a JPG file. I then imported the images into VideoStudio and used the editing software to have each image show as an overlay for set amounts of time. I synchronized title texts to appear with each image to help the viewer to understand what is currently happening with the robot as it executes the program code that was just developed.

I didn't strictly follow my original storyboard and script that I spent hours developing, but instead used it as a guide. Some scenes only deviated a small amount due to imperfect reciting of the script during recording, and other scenes changed a lot because realized while recording that my script just didn't have a natural flow or left something out. I went ahead and filmed everything as I felt it should be and later updated my script and storyboard with the changes.

Neither of my videos stayed within the 5-7 minute time limit, as both are over 8 minutes long. I honestly didn't think it would take a full 7 minutes to accomplish what I'd set out to do, but I realize now that I largely miscalculated my estimate. When given the opportunity to split my video up into two videos, I thought again that each would easily fit in the time limit, but again I was wrong. I thought about cutting each down to 7 minutes to stay within limit, but I just couldn't do so without leaving out something important or having to start over from scratch. Instead, I'm just leaving my videos as slightly too long because I feel that if someone can choose to watch a video for 7 minutes, then they are probably interested enough to watch another minute or two without even realizing it.

For the actual saving of my videos, I chose the Microsoft Windows WMV format as my primary file format. I found that the VideoStudio software can save videos in this file type with very customizable levels of quality while maintaining a decent file size. This was especially crucial for my second video because a large portion of it consists of screen recordings with small, detailed screen text, which required the video to be as clear as possible. VideoStudio can also save the video in different file format, such as .MOV and .AVI, but I wasn't able to achieve as comparable quality and acceptable file size as I was with .WMV. I will still, however, create at least a .MOV file for Apple users, but they may not be able to read the screen text in the second video's screen recorded scenes. Of course, this may not even be an issue because I already uploaded the videos onto YouTube, where they are converted to a format that anyone with a web browser can view. I found the screen recorded text in the second video to be mostly readable, even compressed on YouTube, and I attribute this to the high quality WMV videos that I used as the upload sources.