I made a few changes when working on my CPT, especially with the model. The model’s motor was originally supposed to be in the middle of the model, with the rotating part facing up. It was supposed to turn the rotating part, which would have some sort of bow & arrow attached. I wanted to change it so that the motor would have wheels, and the bow & arrow would be on top, and the motor would move left and right. I wanted it to be like this so it would look better. I then changed it again (because the previous idea was impractical) so that the motor makes some sort of conveyor belt system, to pull on the string.

The program also faced some serious changes. I had originally wanted a simple archery program with a grassy background, targets, and ambient music. I changed all of this because I enjoy more color and brighter music, so I put the same archery program with a mario theme. This also allowed for me to use more sound effects that matched with the theme. My program was also supposed to have a system where the targets would show up for a few seconds then disappear, the user would have to hit the targets before they went away. I got rid of this because each computer processes differently, some are slower than others. Also, the game flowed better with the system I have now, with the 60 second timer and non-changing targets.

One of my problems was an aiming feature I was working on. If the user were to right click, the aim gets smaller, making it more accurate, otherwise it stays big. This feature didn’t work partially because of View.Update and partially because the user would have to hold down the right click to have a smaller aim, not click. Not only that, but the program would not update the view when drawing the bigger circle after letting go of right click, only when the user would move again, somewhat deceiving the user. When i did try to update the view, the target would often cover the aim circle. I simply got rid of this feature because it caused too many problems. Another problem is that I had was that, originally, I had hoped to use the mouse keys to aim. That didn’t work because the motor wouldn’t follow. Not only that, but even if i did break down the program into a bunch of lines separating space, where “if the mouse is in [this] space, move the motor this way”, the program wouldnt know if my mouse is moving left or right. Very early on, I got rid of this feature and used ‘a’ and ‘d’ to move the aim. Another problem I had was that the motor would not move back to its original spot after the game was over. I solved this by using a simple loop that checks if the aim is greater than the middle of the screen or less, and moves left or right accordingly, only stopping when the motor (or aim) is in the middle.

One thing I would change about my program is the lights. I would somehow find a way to make the lights turn on to which target was hit, without disturbing the movement of the motor. I would also add more pictures and colors to my model.