Project Description:

I am creating a pool (billiards) aim trainer with the goal of using computer vision and mathematical modeling to predict where the balls will end up after impact and to help the user get better at playing pool by aiming in better places on the board.

Competitive Analysis:

After looking around, there are some projects which succeed in completing partial aspects of my project but there seem to be none in existence that combine all of the features that this project contains. For example, there is a project titled AR Pool that identifies the path of the ball in real time, but it fails to identify the specific ball and keep track of the balls hit in. In addition, I also want to keep track of the potential for scratching and I failed to find a program that contains that feature. There was another program which recognized the balls and identified them but it did not do path planning to assist the user in aiming properly in order to get the ball(s) in the most optimal position.

Structural Plan:

As of right now, all of the project is in one file which consists of both the gui with tkinter and the OpenCV code. I would like to clean it up and have all of the OpenCV code in separate files which the main gui file will call. In addition, I have organized the images in the project folder that I sourced from pool tournament videos online. As I plan on adding additional features (model for predicting where the balls will go, potential for scratching), I want to keep them in separate files and call them in the gui where they can all be imported and called.

Algorithmic Plan:

I aim to increase the accuracy of the ball identification algorithm in the next few days by gathering a larger dataset to train my identification model on. In addition, I also need to create a ball location prediction algorithm. I plan on doing this by modeling the collisions as mostly elastic collisions in which most of the kinetic energy is transferred. Lastly, I plan on creating the gui with tkinter and will support the user uploading the file and inputting which ball they want to hit.

Timeline:

I hope to have all the algorithms completed at least in their basic state by the TP2 deadline next week. I believe I have the necessary physics background information to write the prediction model and will just need to expand what we learned in class (1d collisions) to the 2d frame. For color recognition, I have the algorithm currently working off the principles of creating a mask and then getting the average color of that location. I then find the smallest difference to the reference value to identify the ball. I believe that I can improve this by implementing some ideas suggested by my mentor.

Version Control: I have google backup and sync installed on both my desktop and laptop which support multiple versions and automatic backup. I have used this for all of my school documents so far and I have not had any problems with recovering files.

Module list: opencv and numpy

Changes for TP2:

I made major changes to the ui taking into feedback from people at Scotland Yard to increase usability. In addition, I was able to get a basic physics model working but I hope to implement monte carlo methods to get a scratch model. The trainer functions now in which the user inputs the image file name and then the simulation will populate the balls based on the identification of the balls that the user uploaded. Then, the user can change the ball they are aiming at, the power they are shooting with and what pocket they are aiming for.