

## Report for Penalty Shootout Game

Philip Kofi Boakye- 13542022 Cohort A

I decided to design a program that can help computer users enjoy a penalty shootout simulation. This game was aimed to help computer users to enjoy their leisure time and also take some stress off their daily work. The game allows the user to press some keys to indicate which direction they intend to shoot and the computer randomly guesses the key the user pressed and attempts to save the ball as well.

This game is a very simply python game that reads and writes to a file and follows objected oriented design. First of all, using the top-down design, my program consisted of these parts: an introduction, the penalty shootout itself and the result component. To ensure the program was designed using the objected oriented approach, I decided to define a class for the penalty shootout game and place all my methods inside it. The game when launched asks the user to select a difficulty mode (using an input function) and once selected it moves on to print the next method. The next method is to print the introduction. With the introduction, the program reads the introduction from a text file called 'Intro' After it returns a well-wishing statement for the user to commence with the game. After, it moves to the "getShootout" method where the actual game occurs. Because we are also implementing the Objected Oriented Design (OOD), we ensure all instance variables have been initialized under the `__init__` method so that they can be accessed at different parts of the program. With the get shootout method, we give the player a list of directions to choose from. We also import the random module to let the computer also randomly select a direction from the list the player chooses from. Using the in-built range function and a for loop, we make sure for all the five penalty kicks whenever the random choice of the computer happens to be same as the user's choice; then the shot has been saved. Otherwise, the user has scored a goal. In the event, the user enters an option outside the range provided, the if statement checks for that and determines the user has entered the wrong key and therefore missed the target. Another exciting feature is the ability of our computer to randomly pick either 3 or 6 out of the nine directions using `random.choices`. This ensures that if your choice is in the range of choices the computer randomly picked, your shot is saved.

At the end of the game, the overall score result of the game is printed and written to a new text file called 'projectresult.' Both the read and write files are closed to ensure that the changes in the files stored on the RAM are effected on the files in the disk.

Also, 'penalties,' which is an instance of the class Pshootout is created as an object and the remaining methods are printed out in the right order. In the getShootout method 'shot' is passed as a parameter since its input is required from the user at that stage and stored under the variable shot.

The written file can be assessed at the end of the shootout to view the final score of the game.