**Extensions:**

* 3D version of Snooker (as a second solution)
* Alternating the different cue sticks, longer shorter or thicker thinner
* Using a slider to control different angles of the cue stick
* Adding a key pressed, 4- randomise coloured balls only, 5 adding in additional coloured balls
* Adding sound effects
* Use of images, audio and video to enhance the UI/UX gameplay

**Note:**

* Comments match with keywords inside rubric
* In video recording, run through the rubric one by one
* Add name, student id, module code module name, date and description in code.
* In video recording, say name and student id, show face is better.
* In video recording, say about your console.
* Inside code, include something like this is mouse pressed function which do what and this is key pressed function which do what.

**To avoid plagiarism:**

* Function name variable name should be changed. Image file name should be change
* Add comments for each single line of code

First, the program runs and without any errors. Program is very easy to use with mouse and key interactions. I have used images for the table and balls to replicate the original snooker table.

[2 points]: Bouncing implemented for the cushions (1 point for matter.js, 1 point for realistic collision)

[2 points]: Bouncing implemented for the balls (1 point for matter.js, 1 point for realistic collision)

[2 points]: Physics implemented for the cue (1 point for matter.js, 1 point for realistic collision)

[1 point]: Balls have the necessary friction

[1 point]: Red balls are removed from the array when in pockets

[2 points]: Cue drawn on screen using mouse and/or key interaction (discuss your choice in report)

[2 points]: Cue manipulated using mouse/key interaction (discuss your choice in report)

[1 point]: Cue has the necessary speed limit

[2 points]: Table starts in three modes. Use key interaction (i.e. keystrokes 1, 2, 3) to load each mode

[2 points]: 1st mode all balls in place as in starting position. Cue ball (white) excluded

[3 points]: 2nd mode use a random algorithm of your choice to get all balls excluding the cue ball (white) on the table. Discuss the choice of the random algorithm in your report

[2 points]: 3rd mode. Adjust the previous step to only randomly allocate the red balls leaving the coloured balls intact. Cue ball is again excluded

[2 points]: Cue ball (white) to be inserted using the necessary constrains and by using mouse and/or key interaction (not random init)

[1 point]: If cue ball falls in the pocket it should be given back to the player and the previous requirement should be executed again

[1 point]: If a coloured balls gets in the pocket it should be returned to its original location

[1 point]: An error prompt is shown when two coloured balls are inserted into the pocket

[3 points]: Collision detection of cue ball

[4 points]: Code presentation: indentation, white space, comments, variable naming

[4 points]: Code competency: code reusability (some functionality is repeated – did student use functions/OO to organise the code better?)

[5 points]: Commentary included? [4 points]: Video included? [14 points]: Has learner implemented any unique ideas for further development?