COMP4471/5472 Project 1, Fall 2015
"**Shark Cage 2D**"

Due date to be announced and posed on course web page
Using WebGL and JavaScript (but not Three.js), and the mathematics package that comes with the textbook, develop a two dimensional interactive game with the following features:

1. The playing field starts as a square centered at the origin with edges orthogonal to coordinate axes. The player is at the origin and facing one of the sides of the square
2. The player stays at the origin, but through interaction using the mouse/and or keyboard can change the facing direction towards any of the 4 sides of the square.
3. This change of direction in (2) must be by way of a smooth rotation taking a finite and discernable time interval
4. The player can shoot in the facing direction
5. A shark appears randomly at one of the sides of the square.
6. The player must readjust the facing direction to face the shark. Meanwhile the shark is chewing part of the side of the square away.
7. A "strength" measure is displayed for each side of the square to indicate how much of the side is chewed away.
8. The player shoots as quickly as possible at the shark and drives it away.
9. A shark then appears randomly at one of the sides.
10. When a side is completely chewed away, the next shark at that side eats the player. (Sharks win.) If the player survives a given number of attacks then player is rescued (player wins)

A well-developed implementation for the above will earn a grade of 85%. To get a higher grade the following should be completed in addition (each feature successfully completed adds 5%)

1. Put a surface feature on sharks (display top view of sharks)
2. Have the shark with the surface feature display some motion.
3. As a side gets more than a set percentage threshold chewed away, the shark at that side scratches the player, and this attracts additional shark(s). For each such side chewed to the threshold, two sharks now randomly appear at different sides (up to a maximum of one at each side)

NOTES:

1. A class demonstration is required for each game (demo on podium machine in classroom)
2. Students may work in teams of two
3. You may need to read ahead for features needed to complete the game.
4. A good approach is to develop the solution in stages, adding features as more background is covered in class.