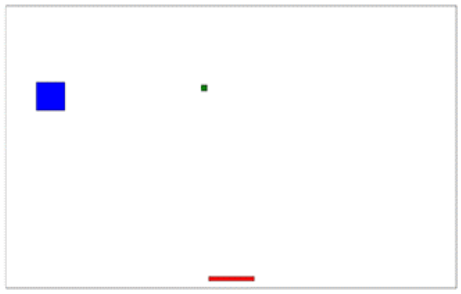
INFO16206: Assignment N2



Evaluation: **10 points**, 10% of your final grade.
Due date: See Dropbox for details

Introduction:

In this assignment, you are expected to demonstrate your knowledge of JavaScript functions, objects, Object-oriented programming and event handling in JavaScript.

Main Requirements:

Please download and import **Assignment2.zip** template project from SLATE.
You **must use** the **CSS file** and **HTML markup** from the template project.
Your application **must work** exactly as shown on the video.

1. You must use template given to you. **You are NOT allowed to modify CSS or HTML code**. You can only modify JavaScript code. You'll get **zero grade** if this requirement not met.
2. Please do NOT change the initial location and speed of the "ball" object.
3. Please do NOT change the initial location of the red "racket" (see "#racket" CSS declarations).
4. Please define "**Racket" constructor function** and create **racket** object (hint: see "ball" object). The speed of the "racket" should be at least 7 px.
5. You must control the "racket" with keyboard keys (left/right arrow keys)**.**
6. The green square "shape" (see "shape" object) must be **randomly placed** somewhere in the **upper right location** of the "container" (**x** coordinate is at least 200px and **y** is less than 250px). The size of the "shape" must be **randomly generated** between 5px and 30px.
7. The "racket", "ball" and "shape" object must always stay within the "container" element. I.e. "racket" should not move outside container and the "ball" must correctly bounce off the "racket".
8. If the "ball" misses the "racket" it should slowly "disappear" at the bottom of the container. The game ends and the red message "You lost" is shown.
9. When the "ball" object "hits" the "shape" the game ends and blue message "You win" is shown.
10. The game ends if you don't win or loose after **20 seconds** of playing. A green message "Draw" must be shown. Make sure this message doesn't appear after you win or loose the game!
11. When the game ends, you must cancel all the events (keyboard, mouse), and clear setInterval() and setTimeout(). You must also "hide" both the "ball" and "shape" elements, but not "racket".
12. The height and weight of the "ball" and "racket" shapes must be determined from the CSS styles (see comments inside **DOMElement** constructor function!).
13. Your code must be reasonably optimized: do not pollute global scope with variables! Create/use methods in the appropriate objects & functions, avoid duplicate code, cache selectors, etc.
14. Make sure there are no console messages of any kind and no debugging code left!
15. JavaScript code must be in the **index.html** file. Do not use external JS files!
16. Do NOT modify **DOMElement** Constructor function! Modify only Ball and Racket functions.

Hints:

1. To randomly place green “shape” you can use **Math.random()** function. See tutorial here: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random
   Don’t forget that “shape” must be placed in the **upper right location** - see requirement N6.

2. Use a constructor function to create a “racket” JavaScript object that inherits properties of the **DOMElement** object (see how it was done for the “ball” object in the template project).

3. To detect “collision” between objects (“ball” and “racket” or “ball” and “shape”) please check **collision()** function in **DOMElement**.
   Check also Axis-Aligned Bounding Box collision detection condition:
   https://developer.mozilla.org/en-US/docs/Games/Techniques/2D_collision_detection

4. You should attach “keydown” event to window object. Check **Exercise9_1**.

5. To hide/show elements you should use **.show()** and **.hide()** methods defined in **DOMElement**.

6. To avoid ball "bouncing" between racket and bottom of the container you need to check the following condition: ball.y + ball.height >= racket.y + racket.height

   The screenshot above shows when this condition happens. It's the moment the game is "lost". You must check this condition before checking collisions!

7. The “ball” can “disappear” at the bottom of the container when you reduce its height slowly at each move until the height becomes less than **ball.y_speed**.

   When ball.y + ball.height >= boundary.height it means "ball" reached the bottom of the container and it's time to reduce is height: this.setDimensions(this.boundary.height - this.y);
   Note: this must be done in combination of Hint N6.

8. To stop the game just call **gameOver()** function, where you’d need to hide objects (ball and shape), remove all event handlers, and clear setInterval() and setTimeout() functions:
   https://www.w3schools.com/jsref/met_element_removeeventlistener.asp
   https://www.w3schools.com/jsref/met_win_clearinterval.asp
   https://www.w3schools.com/jsref/met_win_cleartimeout.asp

9. Check **Comments** in the template and check **Common Mistakes** below!

Common mistakes:

1. You code doesn't run on my computer. This happens when you modify CSS or HTML file(s).

   Remember: JavaScript code must be in the **index.html** file. Do not use external JS files!
   Verify your submission: create a new project using provided template, get the file you've submitted from SLATE and see if your project runs without errors. That's what I'll be doing when grading your assignment.

2. Ball does not slowly disappear at the bottom – see Hint N7.

3. Ball is bouncing between the floor and the racket. It happens when you move racket on top of the "ball" before it hits the bottom of the container. Change the speed of the "ball" in setInterval() and try it with slow moving "ball".
   Solution: check "collision" between "racket" AND "ball" after condition in Hint N6.

4. Hardcoded values used.
   You must determine the size of the objects from CSS (for instance, when you create "Racket" or "Ball" objects).
   Verify your code: change height/weight of the "ball" in CSS, change "top", "left" and "width" CSS properties of the "racket". Your code must work with these changes!

5. Many functions used.
   Solution: set all the properties and methods inside corresponding objects! Except for random number generator there should be no more global functions declared beyond what's given in the template!

6. Keyboard event(s) not cancelled when game ends or some shapes still shown.
   Solution: see Hint N5 and Hint N8. Make sure "racket" doesn't move when game is over!

7. Global variables used.
   Just like mistake N4, no need to have global variables beyond three major ones: "shape", "ball" and "racket". You might use global **timeoutID** to show "Draw" message. Ideally it should be set in the "container" object though!

Submission Requirements:

- Please save index.html file as a text file **index.txt**
- You must upload only **index.txt** file to the Assignment Dropbox by the Due Date.
  You'll receive **zero grade** if you submit **index.html** file or file with a different name or extension.
- Your submission must be unique or have references.
- Please self-evaluate your code in the comments section of the Dropbox.
- **Late submissions are penalized 10% / day and 0 after 3 days!**