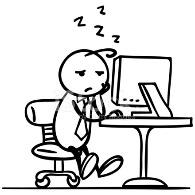
We have gone through some of the basic functionality of jQuery, now it’s time for us to have some hands-on experience!

Before delving deep into the project, let’s briefly discuss how HTML5, CSS and JavaScript (jQuery) become the main stream for web development.

Back in year 1996, Adobe Flash was introduced and it quickly became a popular method for adding animation and interactivity to web pages. Developers used it to create interactive webpages and browser-based game which is also known as Flash game. With the launch of HTML5, mobile has emerged as one of the publishing world’s biggest advantages. HTML5, CSS and JavaScript are widely adopted by the major web browser and are compatible with touch screens. Flash on the other hand, was designed for desktop, it consumes more power causes the device to heat up easily. Hence developers quickly adopted HTML5, CSS and JavaScript taking advantage of their efficiency and cross-platform compatibility. Today, although Flash websites still exist, they are mostly in the process of shifting towards the modern web technologies. A very good example will be Scratch, a block programming platform to learn programming, is launching a Beta version for Scratch 3.0 which is built upon the modern web technologies and will be fully releasing the latest version of Scratch in January 2019. Unlike its predecessor, Scratch 3.0 does not require Flash player plugins and is designed for touch-screen devices.

After the lengthy and boring historical explanation, now it is time for us to jump right into the project which you must have been waiting for.

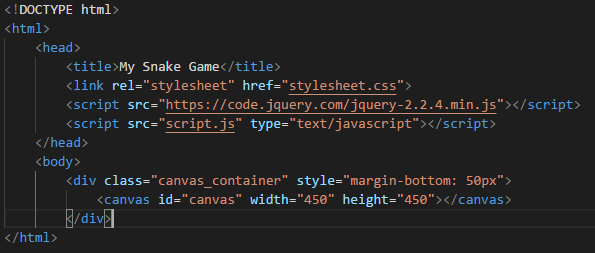


We are going to take advantage of jQuery cross-platform compatibility, to create our very own Snake Game, which is not only **playable on** a computer but also on your **TOUCH-SCREEN MOBILE DEVICES!!!**

**Hands-on Project: Snake Game**

Download and unzip the folder [], go to the “Snake Game”, you should be able to see 3 files and 1 folder namely “snake.html”, “stylesheet.css”, “script.js” and “snake\_arrows”.

Open the snake.html with your favorite text editor and you will see the following code.



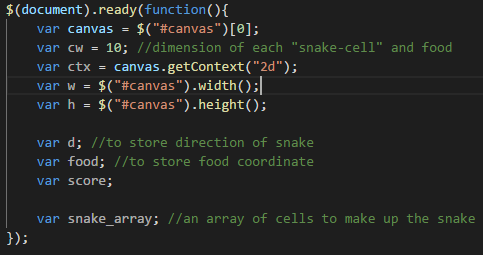
In the <head> tag, we included jQuery, and our own CSS and JavaScript file. Meanwhile in the <body> tag, we have a container containing **canvas** object. Canvas is the place where our game will go.

In our own CSS file, stylesheet.css, will be some size and positioning definition of the DOM in our html. (If you would like to know more, please refer to this awesome website <https://www.w3schools.com/css/>)

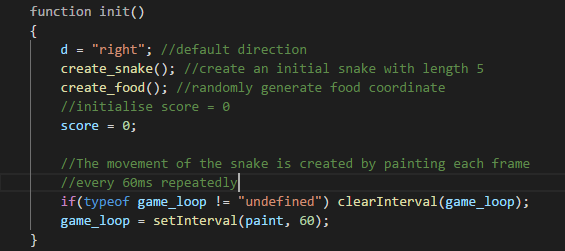
Now open style.js with your favorite text editor, we start a jQuery program with the following line:

This document ready event prevents any jQuery code from running before the webpage is fully loaded.

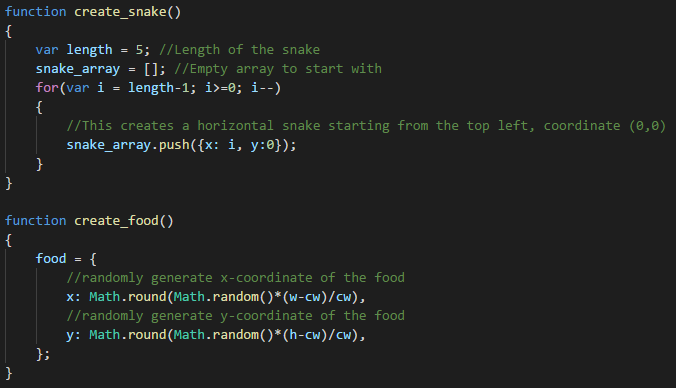
Add the following lines.



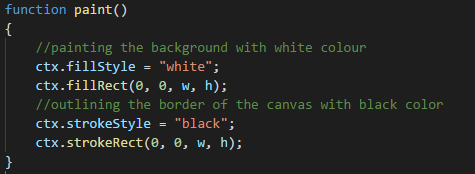
Here we declared some variable for later use. The variable canvas is a variable that store the canvas DOM in our html file. The ctx variable on the other hand contains the context identifier defining the drawing context associated to the canvas. Here we use jQuery id selector $(“#canvas”) to select the html tag with id = “canvas”.



Then we add a function init() which will initialize a new game whenever we start the game or game is over. The movement of the snake is created by painting frame by frame every 60ms.



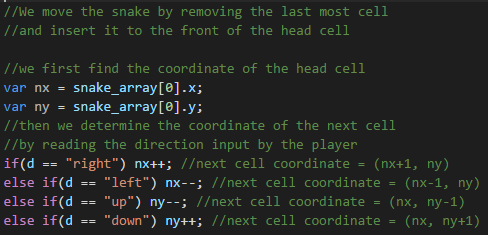
We now create two function, where create\_snake will generate a new snake of length 5 whereas create\_food will generate new coordinate for the food.



We will then work on the function paint, which will be called every 60ms. We start by painting the background white and outlining the canvas.

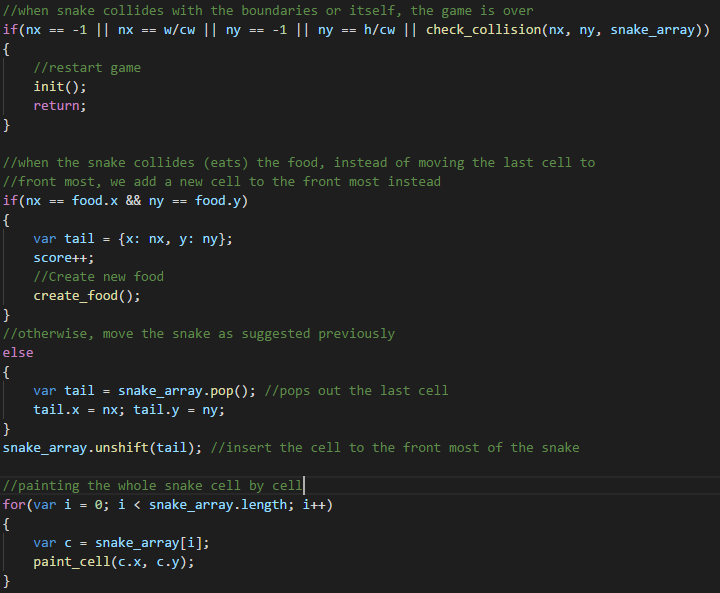
Next we will be looking into how we are going to move the snake.

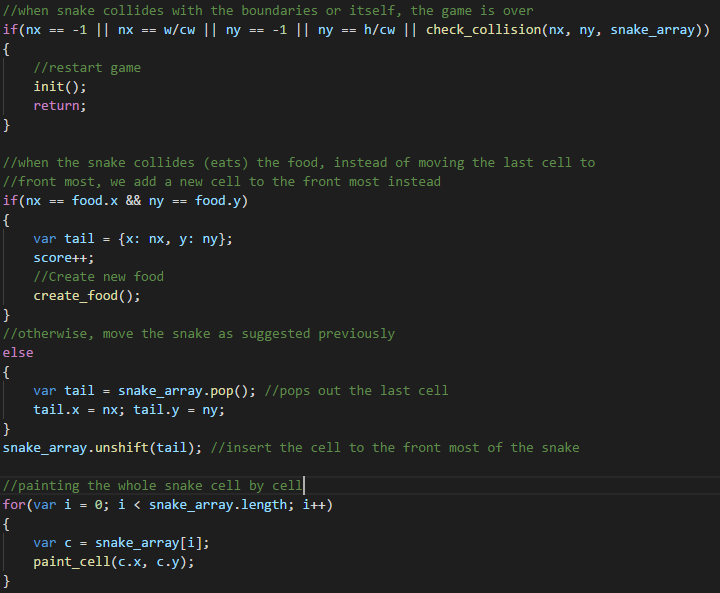
We move the snake simply by taking the last cell of snake and move it to the front of that current head cell. Add the following code to paint() function.



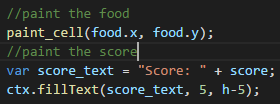
Here, we determine the coordinate of where the snake should head from the user key input.

We will now define how the snake should behave when it collides with boundaries or itself and when the snake eats the food. Continue by adding the following code to the paint function.

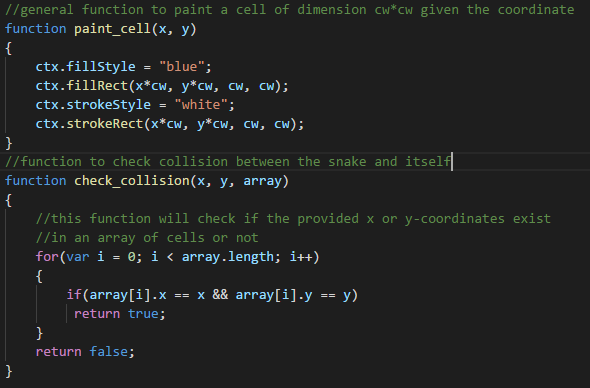




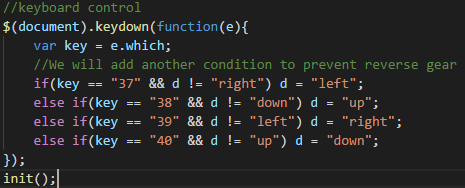
When the snake collides with boundaries or itself, game is over and a new game is initiated. The length of the snake is increased by one when it eats the food. Then we paint out the whole snake cell by cell.



We end the paint() function by painting the food and scores on the canvas.



We then add two new functions which will paint a new cell given the coordinate and the other will check for collision of the snake with itself.



Last but not least we will add the keyboard control to the snake. The function will be triggered when a keyboard key is pressed down, then the function will check for the key which is being pressed down, if the key is one of the arrow keys, corresponding action will be taken by changing the direction of d. We then call the init() function to start the game.

We are now done with the code. Now let’s try to run our very own snake game created with HTML and jQuery.

To run our code, right click on the snake.html, and open it with you preferred browser.

Tadaa!!! You have successfully your Snake Game that can be played on a browser.

