**Assignment 7`**

Your task is to implement the virtual pet app that you started working on in assignment # 3 using JavaScript and**jQuery**.  This week you’ll implement the user interface.

The following screencast shows a simple implementation of the app.  Your implementation has to implement the same logic.



When the web page is loaded, the user has the option to adopt a pet.  The user should be able to choose between a dog and a fish by clicking on the corresponding image. You may implement more choices if you like.

Once the user adopts a pet, the actions required to take care of that pet should appear as buttons:  for a dog, the buttons should include feeding, walking and playing and for a fish the buttons should be feeding and cleaning.  Walking the fish should of course not be an option.

The program should also keep track of time.  This can start after the adoption.  The program should invoke the newDay() method on the pet object repeatedly.  For testing purposes, we’ll assume that**the day lasts 60 seconds in our virtual world**.  The user should also 'see' the time change: one way to do that is by making the background color change (day to night to day…)

The program should also invoke the check() method on the pet object when needed and signal that the pet needs attention.  We’ll assume that pets become agitated when they need attention.  In the screencast, you’ll see that the pet starts shaking.  When all the needs of the pet are met, the pet stops shaking.

The html source document  and css stylesheet that were used in the screencast demo are available under Resources.  A JavaScript**template file pet.js** file is also available under Resources.  It contains a modified solution to assignment 3.  The check method has been modified to return true if the pet needs attention and false otherwise.  The pet name is still defined but you are not required to use it in this assignment.  Note that because we are implementing the user interface, we don’t need to use console.log to signal that the pet needs attention. **Do NOT change anything in the object oriented implementation**included in the template.

To submit your assignment, upload your**JavaScript program, your html source document and your css stylesheet**.

**Start early, ask questions and have fun!**

**Grading Rubric:**

The program correctly uses jQuery (not DOM methods) wherever possible**-**10 points

The web page looks right after dog is adopted – 5 points

The web page looks right after fish is adopted – 5 points

The action buttons (feed, play, walk, clean) all work correctly and reset the corresponding property in the pet object - 16 points

newDay() is invoked on the pet object every 60 seconds and the boolean pet properties are all reinitialized -10 points

The pet becomes agitated whenever it needs attention - 7 points

The pet stops moving when all its needs are met - 7 points

The passage of time is signaled somehow (background color change or some other way) - 10 points

**Answer**

* text/css[pet.css](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/15080502/a279cc8f-b3db-421f-800f-da7ced315a0a/pet.css)
* application/x-javascript[pet.js](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/15080502/0c054d76-358b-4a70-8081-8e35232735e1/pet.js)
* text/html[virtualpet.html](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/15080502/f2a55f5c-787c-4342-009c-8afa2cc5a3ad/virtualpet.html)
* application/x-javascript[jquery-1.11.1.js](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/15080502/599efa9e-63ad-4bc2-8043-3ba7c29da355/jquery-1.11.1.js)

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/15080502/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/15080502/list)

/\*

 \* Virtual Pet App with User Interface - Solution

 \*

 \*/

$(document).ready(function () {

    'use strict';

    // Here's the object oriented model implementation from a previous assignment

    // DO NOT MODIFY THIS PART

    // Create the pet protoptype

    var pet = {

        hungry: true,

        ill: false,

        name: 'Your Pet'

    };

    pet.feed = function () {

        // The pet is no longer hungry.

        // Return a string indicating that the pet is full.

        this.hungry = false;

        return this.name + ' is full.';

    };

    pet.newDay = function () {

        // Set ALL the boolean properties to true.

        // Return 'Good morning!'

        for (var prop in this) {

            if (typeof this[prop] === 'boolean') {

                this[prop] = true;

            }

        }

        return 'Good morning!';

    };

    pet.check = function () {

        // Check ALL the boolean properties of the pet object.

        // Return true if the pet needs attention.

        // Return false is the pet is fine.

        var result = false;

        for (var prop in this) {

            if (this[prop] === true) {

                result = true;

            }

        }

        return result;

    };

    // Create the fish prototype.

    var fish = Object.create(pet);

    fish.clean = function () {

        // Set the object unhealthy property to false.

        // Return a string of the form: 'pet name likes the clean tank.'

        this.ill = false;

        return this.name + ' likes the clean tank.';

    };

    // Create the dog prototype.

    var dog = Object.create(pet);

    // initialize the lonely property

    dog.lonely = false;

    dog.walk = function () {

        // Set the object's unhealthy property to false.

        // Return a string of the form: 'pet name enjoyed the walk!'

        this.ill = false;

        return this.name + ' enjoyed the walk!';

    };

    dog.play = function () {

        // Set the object's lonely property to false.

        // Return a string of the form:  'pet name loves you.'

        this.lonely = false;

        return this.name + ' loves you.';

    };

/\*---------------------------------------------------------------------------------------------\*/

    // Do not modify anything above this line

    // New code starts here

    // These are the User Interface functions - View

    function updateUI(species) {

        // Rearrange the images and the buttons to reflect the specific adoption

        $('#adopt').text('');  // Clear the text of the header

        $('#feed').show();

        if (species === 'dog') {

            $('#fish').hide();

            $('#walk').show();

            $('#play').show();

        } else if (species === 'fish') {

            $('#dog').hide();

            $('#clean').show();

        }

    }

    function updatePetUI(petObject, petSpecies) {

        // Check to see if the pet needs attention

        // Update the UI accordingly to show that.

        var imageSelector = '#' + petSpecies;

        if (petObject.check()) {

            // pet needs attention

            $(imageSelector).addClass('move');

        } else {

            $(imageSelector).removeClass('move');

        }

    }

    function changeUIBackground() {

        // Initiate a background change - from day to night and vice-versa

        $('body').toggleClass('day night');

        }

    // This is the main function - it controls the app logic

    function adopt(event) {

        var myPet;    // this is the adopted pet.

        var species = event.target.id; // the species of the adopted pet:  fish or dog?

        // Update the data model to reflect the adoption

        if (species === 'dog') {

            myPet = Object.create(dog);

        } else if (species === 'fish') {

            myPet = Object.create(fish);

        } else {  // enable future enhancements

            myPet = Object.create(pet);

        }

        // disable further adoptions

        $('#choice').off('click');

        // Upate the UI

        updateUI(species);

        updatePetUI(myPet, species)

        // add an event listener for the action buttons.

        $('#action').click(function(event) {

            // process the action initiated by the corresponding button

            // the event target here is one of the action buttons.

            if (event.target.id === 'feed') {

                myPet.feed();

            } else if (event.target.id === 'play') {

                myPet.play();

            } else if (event.target.id === 'walk') {

                myPet.walk();

            } else if (event.target.id === 'clean') {

                myPet.clean();

            };

            updatePetUI(myPet, species)  // Update the pet image if needed

        });

        // Schedule a background change - every 30 seconds

        setInterval(changeUIBackground, 30000);

        // Invoke newDay every 60 seconds

        setInterval(function() {

            myPet.newDay();

            //update the pet image if needed

            updatePetUI(myPet, species)

        }

                , 60000);

    }

    // main program starts here

    // Register pet adoption event handler

    $('#choice').click(adopt);

});

**Comments**

Excellent!

One minor issue:

After the adoption, the message 'Please adopt a pet by clicking on an image below:' should no longer be displayed.

Late submission.

 1 of 1