**Documentatie Lab4 JQuery : Petrut Adrian-Damian, gr226**

**Problema 1:** In fisierul html am 2 taguri select.Dupa incarcarea documentului ($function(){}) selectez fiecare lista folosind selectorul jQuery $(“#id”) si atasez un eventHandler pt evenimentul dblClick. Functia “move” obtine elementul selectat folosind “.find(“:selected”)” si il adauga in cealalta lista folosind appendTo.

**Problema 2:** In fisierul html am un form ce contine inputurile necesare si un input de submit. Diferente fata de versiunea pur javascript: getElementById inlocuit cu $(“#id”). Pentru a seta prorietati css se foloseste metoda jQuery css(). Pentru a extrage valoarea din inputuri se foloseste apelul metodei val(), al obiectelor wrapper din jquery.

**Problema 3:** In fisierul html exista un div cu #id “container” inauntrul caruia se vor adauga divurile corespunzatoare fiecarui element din memory game. In versiunea JQuery div-urile corespunzatoare fiecarui element sunt atasate folosind functia append. Valoarea elementelor nu mai este verificata folosind vectorul identity(acesta este folosit doar pentru shuffle-ul initial), ci este verificata folosind direct wrapper-ul spanului corespunzator. Vectorul flipped acum contine obiecte jQuery, se seteaza, folosind metoda jQuery attr un atribut “valid” care indica daca se poate da click sau nu pe un element. Vizibilitatea se schimba folosind metoda css a obiectelor jQuery. Functia createMemoryGame este inlocuita cu o functie anonima ce se apeleaza dupa incarcarea paginii.

**Problema 4:** In fisierul html am 4 tabele, dintre care 2 au headerele pe prima linie , iar 2 le au pe prima coloana. In versiunea jQuery apelez, dupa incarcarea paginii, pentru fiecare tabel cu clasa orizontal functia attachSortEventHorizontalHeaderTable, folosind metoda jQuery each. Aceasta functie ataseaza event handlere pentru click pe fiecare dintre th-urile tabelului. Starea de sortare (crescator,descrescator)nu mai este memorata intr-un dictionar , ci o adaug ca atribut pe headere folosind metoda jQuery attr. Pentru a verifica celula de la un anumit index folosesc selectorul jQuery “eq”. Pentru a schimba ordinea liniilor folosesc metoda jQuery before. Analog pentru tabelul cu header vertical.

**Problema 5:** In fisierul html am o lista neordonata. Fiecare item din lista contine o imagine si un link.Pentru a aseza linkul peste imagine setez position:absolute si pozitionez corespuzator. Elementele listei au pozitionare absoluta si sunt toate pozitionate in acelasi loc(sus stanga). Initial setez proprietatea diplay la none pentru toate exceptand prima. Pentru a face tranzitia la urmatoarea imagine apelez functia fadeIn si fadeOut. Functia fadeIn va schimba display-ul elementului la block, si ii va schimba gradual opacitatea de la 0 la 1. Functia fadeOut va schimba opacitatea pana la 0,la finalul ei se va seta display :none (automat). Astfel pentru a face tranzitia de la imaginea x la imaginea x+1, apelez in acelasi timp ( folosind setTimeout) fadeOut pe imaginea x si fadeIn pe imaginea x+1. De asemenea la clickul pe buton acesta va fi disabled pana se incarca urmatoarea imagine. Obtin acest lucru atasand un callback function pentru functia fadeOut, acest callback va fi apelat la finalul ei. In el readuc butoanele la starea de enabled: “$(“button”).prop(“disabled”,false);

**Problema 6:** Vectorul identity retine valorile elementelor. Liniile si celulele tabelului sunt create dinamic folosind jQuery ($(“<td></td>”) pt celula), iar apoi adaugate folosind metoda append(). Pentru a raspunde la apasarea sagetilor atasez un event handle $(document).keydown(function(e){..}). e.keyCode va fi codul asociat tastei.

Actiunile se trateaza astfel: o mutare catre dreapta a casutei libere corespunde unui switch(array[x],array[x+1]), o mutare in jos corespunde unui switch(array[x],array[x+n]), unde n e numarul de elemente pe o linie, analog pentru stanga si sus.