МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ІСМ

Звіт

з лабораторної роботи №4

з дисципліни «Екстремальне програмування»

Виконав

Студент групи КН-311

Лапінський О.В

Перевірив

Викладач

Щербак С.С

Львів 2020

**Завдання:** розробити проект з використанням однієї з запропонованих мов програмування, користуючись методикою Екстримального програмування.

В якості мови програмування було обрано JavaScript.

Даний проект представляє собою найпростіший вид тестів коли набір результатів потрібно співставити за певними ознаками.

В даній лабораторній роботі буде реалізовано, JS код для працездатності тесту

Продовження основного функціоналу JS скрипта:

/\* VARIABLES YOU COULD MODIFY \*/  
var boxSizeArray = [3,3,3,3,3,3,3]; // Array indicating how many items there is rooom for in the right column ULs  
  
var arrow\_offsetX = -5; // Offset X - position of small arrow  
var arrow\_offsetY = 0; // Offset Y - position of small arrow  
  
var arrow\_offsetX\_firefox = -6; // Firefox - offset X small arrow  
var arrow\_offsetY\_firefox = -13; // Firefox - offset Y small arrow  
  
var verticalSpaceBetweenListItems = 3; // Pixels space between one <li> and next  
// Same value or higher as margin bottom in CSS for #dhtmlgoodies\_dragDropContainer ul li,#dragContent li  
  
  
var initShuffleItems = true; // Shuffle items before staring  
  
var indicateDestionationByUseOfArrow = true; // Display arrow to indicate where object will be dropped(false = use rectangle)  
  
  
var lockedAfterDrag = true; /\* Lock items after they have been dragged, i.e. the user get's only one shot for the correct answer \*/  
  
  
/\* END VARIABLES YOU COULD MODIFY \*/  
  
var dragDropTopContainer = false;  
var dragTimer = -1;  
var dragContentObj = false;  
var contentToBeDragged = false; // Reference to dragged <li>  
var contentToBeDragged\_src = false; // Reference to parent of <li> before drag started  
var contentToBeDragged\_next = false; // Reference to next sibling of <li> to be dragged  
var destinationObj = false; // Reference to <UL> or <LI> where element is dropped.  
var dragDropIndicator = false; // Reference to small arrow indicating where items will be dropped  
var ulPositionArray = new Array();  
var mouseoverObj = false; // Reference to highlighted DIV  
  
var MSIE = navigator.userAgent.indexOf('MSIE')>=0?true:false;  
var navigatorVersion = navigator.appVersion.replace(/.\*?MSIE (\d\.\d).\*/g,'$1')/1;  
var destinationBoxes = new Array();  
var indicateDestinationBox = false;  
  
function getTopPos(inputObj)  
{  
 var returnValue = inputObj.offsetTop;  
 while((inputObj = inputObj.offsetParent) != null){  
 if(inputObj.tagName!='HTML')returnValue += inputObj.offsetTop;  
 }  
 return returnValue;  
}  
  
function getLeftPos(inputObj)  
{  
 var returnValue = inputObj.offsetLeft;  
 while((inputObj = inputObj.offsetParent) != null){  
 if(inputObj.tagName!='HTML')returnValue += inputObj.offsetLeft;  
 }  
 return returnValue;  
}  
  
function cancelEvent()  
{  
 return false;  
}  
function initDrag(e) // Mouse button is pressed down  
{  
 if(document.all)e = event;  
 if(lockedAfterDrag && this.parentNode.id!='allItems')return;  
 var st = Math.max(document.body.scrollTop,document.documentElement.scrollTop);  
 var sl = Math.max(document.body.scrollLeft,document.documentElement.scrollLeft);  
  
  
  
 dragTimer = 0;  
 dragContentObj.style.left = e.clientX + sl + 'px';  
 dragContentObj.style.top = e.clientY + st + 'px';  
 contentToBeDragged = this;  
 contentToBeDragged\_src = this.parentNode;  
 contentToBeDragged\_next = false;  
 if(this.nextSibling){  
 contentToBeDragged\_next = this.nextSibling;  
 if(!this.tagName && contentToBeDragged\_next.nextSibling)contentToBeDragged\_next = contentToBeDragged\_next.nextSibling;  
 }  
 timerDrag();  
 return false;  
}  
  
function everythingIsCorrect()  
{  
 alert('Congratulations! Everything is correct');  
}  
  
  
function timerDrag()  
{  
 if(dragTimer>=0 && dragTimer<10){  
 dragTimer++;  
 setTimeout('timerDrag()',10);  
 return;  
 }  
 if(dragTimer==10){  
 dragContentObj.style.display='block';  
 dragContentObj.appendChild(contentToBeDragged);  
 }  
}  
  
function moveDragContent(e)  
{  
 if(dragTimer<10){  
 if(contentToBeDragged){  
 if(contentToBeDragged\_next){  
 contentToBeDragged\_src.insertBefore(contentToBeDragged,contentToBeDragged\_next);  
 }else{  
 contentToBeDragged\_src.appendChild(contentToBeDragged);  
 }  
 }  
 return;  
 }  
 if(document.all)e = event;  
 var st = Math.max(document.body.scrollTop,document.documentElement.scrollTop);  
 var sl = Math.max(document.body.scrollLeft,document.documentElement.scrollLeft);  
  
  
 dragContentObj.style.left = e.clientX + sl + 'px';  
 dragContentObj.style.top = e.clientY + st + 'px';  
  
 if(mouseoverObj)mouseoverObj.className='';  
 destinationObj = false;  
 dragDropIndicator.style.display='none';  
 if(indicateDestinationBox)indicateDestinationBox.style.display='none';  
 var x = e.clientX + sl;  
 var y = e.clientY + st;  
 var width = dragContentObj.offsetWidth;  
 var height = dragContentObj.offsetHeight;  
  
 var tmpOffsetX = arrow\_offsetX;  
 var tmpOffsetY = arrow\_offsetY;  
 if(!document.all){  
 tmpOffsetX = arrow\_offsetX\_firefox;  
 tmpOffsetY = arrow\_offsetY\_firefox;  
 }  
  
 for(var no=0;no<ulPositionArray.length;no++){  
 var ul\_leftPos = ulPositionArray[no]['left'];  
 var ul\_topPos = ulPositionArray[no]['top'];  
 var ul\_height = ulPositionArray[no]['height'];  
 var ul\_width = ulPositionArray[no]['width'];  
  
 if((x+width) > ul\_leftPos && x<(ul\_leftPos + ul\_width) && (y+height)> ul\_topPos && y<(ul\_topPos + ul\_height)){  
 var noExisting = ulPositionArray[no]['obj'].getElementsByTagName('LI').length;  
 if(indicateDestinationBox && indicateDestinationBox.parentNode==ulPositionArray[no]['obj'])noExisting--;  
 if(noExisting<boxSizeArray[no-1] || no==0){  
 dragDropIndicator.style.left = ul\_leftPos + tmpOffsetX + 'px';  
 var subLi = ulPositionArray[no]['obj'].getElementsByTagName('LI');  
 for(var liIndex=0;liIndex<subLi.length;liIndex++){  
 var tmpTop = getTopPos(subLi[liIndex]);  
 if(!indicateDestionationByUseOfArrow){  
 if(y<tmpTop){  
 destinationObj = subLi[liIndex];  
 indicateDestinationBox.style.display='block';  
 subLi[liIndex].parentNode.insertBefore(indicateDestinationBox,subLi[liIndex]);  
 break;  
 }  
 }else{  
 if(y<tmpTop){  
 destinationObj = subLi[liIndex];  
 dragDropIndicator.style.top = tmpTop + tmpOffsetY - Math.round(dragDropIndicator.clientHeight/2) + 'px';  
 dragDropIndicator.style.display='block';  
 break;  
 }  
 }  
 }  
  
 if(!indicateDestionationByUseOfArrow){  
 if(indicateDestinationBox.style.display=='none'){  
 indicateDestinationBox.style.display='block';  
 ulPositionArray[no]['obj'].appendChild(indicateDestinationBox);  
 }  
  
 }else{  
 if(subLi.length>0 && dragDropIndicator.style.display=='none'){  
 dragDropIndicator.style.top = getTopPos(subLi[subLi.length-1]) + subLi[subLi.length-1].offsetHeight + tmpOffsetY + 'px';  
 dragDropIndicator.style.display='block';  
 }  
 if(subLi.length==0){  
 dragDropIndicator.style.top = ul\_topPos + arrow\_offsetY + 'px'  
 dragDropIndicator.style.display='block';  
 }  
 }  
  
 if(!destinationObj)destinationObj = ulPositionArray[no]['obj'];  
 mouseoverObj = ulPositionArray[no]['obj'].parentNode;  
 mouseoverObj.className='mouseover';  
 return;  
 }  
 }  
 }  
}  
  
function checkAnswers()  
{  
  
 for(var no=0;no<destinationBoxes.length;no++){  
 var subLis = destinationBoxes[no].getElementsByTagName('LI');  
 if(subLis.length<boxSizeArray[no])return;  
  
 for(var no2=0;no2<subLis.length;no2++){  
 if(subLis[no2].className=='wrongAnswer')return;  
 }  
 }  
  
 everythingIsCorrect();  
  
  
}  
  
  
/\* End dragging  
Put <LI> into a destination or back to where it came from.  
\*/  
function dragDropEnd(e)  
{  
 if(dragTimer==-1)return;  
 if(dragTimer<10){  
 dragTimer = -1;  
 return;  
 }  
 dragTimer = -1;  
 if(document.all)e = event;  
 if(destinationObj){  
 var groupId = contentToBeDragged.getAttribute('groupId');  
 if(!groupId)groupId = contentToBeDragged.groupId;  
  
 var destinationToCheckOn = destinationObj;  
 if(destinationObj.tagName!='UL'){  
 destinationToCheckOn = destinationObj.parentNode;  
 }  
  
 var answerCheck = false;  
 if(groupId == destinationToCheckOn.id){  
 contentToBeDragged.className = 'correctAnswer';  
 answerCheck=true;  
 }else{  
 contentToBeDragged.className = 'wrongAnswer';  
 }  
 if(destinationObj.id=='allItems' || destinationObj.parentNode.id=='allItems')contentToBeDragged.className='';  
  
  
 if(destinationObj.tagName=='UL'){  
 destinationObj.appendChild(contentToBeDragged);  
 }else{  
 destinationObj.parentNode.insertBefore(contentToBeDragged,destinationObj);  
 }  
 mouseoverObj.className='';  
 destinationObj = false;  
 dragDropIndicator.style.display='none';  
 if(indicateDestinationBox){  
 indicateDestinationBox.style.display='none';  
 document.body.appendChild(indicateDestinationBox);  
 }  
  
 contentToBeDragged = false;  
  
 if(answerCheck)checkAnswers();  
  
 return;  
 }  
 if(contentToBeDragged\_next){  
 contentToBeDragged\_src.insertBefore(contentToBeDragged,contentToBeDragged\_next);  
 }else{  
 contentToBeDragged\_src.appendChild(contentToBeDragged);  
 }  
 contentToBeDragged = false;  
 dragDropIndicator.style.display='none';  
 if(indicateDestinationBox){  
 indicateDestinationBox.style.display='none';  
 document.body.appendChild(indicateDestinationBox);  
  
 }  
 mouseoverObj = false;  
  
}  
  
/\*  
Preparing data to be saved  
\*/  
function saveDragDropNodes()  
{  
 var saveString = "";  
 var uls = dragDropTopContainer.getElementsByTagName('UL');  
 for(var no=0;no<uls.length;no++){ // LOoping through all <ul>  
 var lis = uls[no].getElementsByTagName('LI');  
 for(var no2=0;no2<lis.length;no2++){  
 if(saveString.length>0)saveString = saveString + ";";  
 saveString = saveString + uls[no].id + '|' + lis[no2].id;  
 }  
 }  
  
 document.getElementById('saveContent').innerHTML = '<h1>Ready to save these nodes:</h1> ' + saveString.replace(/;/g,';<br>') + '<p>Format: ID of ul |(pipe) ID of li;(semicolon)</p><p>You can put these values into a hidden form fields, post it to the server and explode the submitted value there</p>';  
  
}

Даний скрипт містить основні змінні і частину функціоналу для роботи тесту.

Висновок: На даній роботі було додано частину JS скрипта для роботи майбутньої сторінки.