(function($) {

var nonMatchingCardTime = 1000;

  //كام صورة عندنا عشان نعمل كروت علي اساسها

  var imagesAvailable = 15;

**انونيماس فانكشن بتعمل حبشتكنات كده بس بتخلي الكود كله يتنفذ مرة واحدة جوا نفس الاسكوب بس منقدرش نقرأها من بره**

**تشغيل السيتنج من علامة الترس**

var settings = document.getElementById('memory--settings-icon');

  var modal = document.getElementById('memory--settings-modal');

  var handleOpenSettings = function (event) {

    event.preventDefault();

    modal.classList.toggle('show');

  };

  settings.addEventListener('click', handleOpenSettings);

**الفاكشن اللي بتخلي زرار ستارت يشغل اللعبة**

// التعامل مع زرار الاستارت بيعمل ريسارت لكل حاجه

  var reset = document.getElementById('memory--settings-reset');

  var handleSettingsSubmission = function (event) {

    event.preventDefault();

//اسحب القيم بتاعت الصفوف و الاعمدة و اتأكد منها من فاكشن البداية في الكونترولر

var selectWidget = document.getElementById("memory--settings-grid").valueOf();

   var grid = selectWidget.options[selectWidget.selectedIndex].value;

   var gridValues = grid.split('x'); //تقسم "2x3" to {2,3}

   var cards = $.initialize(Number(gridValues[0]), Number(gridValues[1]), imagesAvailable);

// بيمسح الكلاسات بتاعت العرض و يفضي القيم القديمة و يشغل فاكشن عرض الواجهة

    if (cards) {

      document.getElementById('memory--settings-modal').classList.remove('show');

      document.getElementById('memory--end-game-modal').classList.remove('show');

      document.getElementById('memory--end-game-message').innerText = "";

      document.getElementById('memory--end-game-score').innerText = "";

      buildLayout($.cards, $.settings.rows, $.settings.columns);

    }

**بيشغل الهبد اللي فوق ده لما تكليك علي الزرار بتاع ستارت**

reset.addEventListener('click', handleSettingsSubmission);

**اللي بيحصل لما تضغط علي الكارت**

// Handle clicking on card

  var handleFlipCard = function (event) {

    event.preventDefault();

    var status = $.play(this.index);

    console.log(status);

    if (status.code != 0 ) {

      this.classList.toggle('clicked');

    }

// لو الكود ب1 او 2 اعمل الكلاس بكليكيد

    if (status.code == 3 ) {

      setTimeout(function () {

        var childNodes = document.getElementById('memory--cards').childNodes;

        childNodes[status.args[0]].classList.remove('clicked');

        childNodes[status.args[1]].classList.remove('clicked');

      }.bind(status), nonMatchingCardTime);

    }

// لو الكود ب3 يبقي القيم مش شبه بعض و صفرها

    else if (status.code == 4) {

      var score = parseInt((($.attempts - $.mistakes) / $.attempts) \* 100, 10);

      var message = getEndGameMessage(score);

      document.getElementById('memory--end-game-message').textContent = message;

      document.getElementById('memory--end-game-moves').textContent = 'Moves: ' + $.attempts;

      document.getElementById('memory--end-game-mistakes').textContent = 'Mistakes: ' + $.mistakes;

      document.getElementById('memory--end-game-score').textContent = 'Score: ' + score + ' / 100';

      document.getElementById("memory--end-game-modal").classList.toggle('show');

    }

// لو الكود ب4 يبقي اللعبة خلصت و صفر القيم و اظهر الاسكور

  };

**تحديد الرسالة حسب الاسكور**

var getEndGameMessage = function(score) {

    var message = "";

    if (score == 100) {

      message = "Excellent job!"

    }

    else if (score >= 70 ) {

      message = "Great job!"

    }

    else if (score >= 50) {

      message = "Good job!"

    }

    else {

      message = "You can do better.";

    }

    return message;

  }

**الفاكشن اللي بتعرض الواجهة بتاعت الكروت و تتحكم في طول و عرض كل كارت**

var buildLayout = function (cards, rows, columns) {

    if (!cards.length) {

      return;

    }

    var memoryCards = document.getElementById("memory--cards");

    var index = 0;

    var cardMaxWidth = document.getElementById('memory--app-container').offsetWidth / columns;

    var cardHeightForMaxWidth = cardMaxWidth \* (3 / 4);

    var cardMaxHeight = document.getElementById('memory--app-container').offsetHeight / rows;

    var cardWidthForMaxHeight = cardMaxHeight \* (4 / 3);

    //بيسحب قيمة الشاشة و يخلي الطول و العرض ثابت حسب كل شاشة

    // بيمسح كل الداتا القديمة في الصورة اول قبل ما يعمل اي حاجه

    while (memoryCards.firstChild) {

      memoryCards.firstChild.removeEventListener('click', handleFlipCard);

      memoryCards.removeChild(memoryCards.firstChild);

    }

    for (var i = 0; i < rows; i++) {

      for (var j = 0; j < columns; j++) {

        // بيزود عدد الكروت حسب عدد الصفوف و الاعمدة

        memoryCards.appendChild(buildCardNode(index, cards[index],

          (100 / columns) + "%", (100 / rows) + "%"));

        index++;

      }

    }

    // تغيير حجم الكروت عشان تتناسب مع حجم الشاشة

    if (cardMaxHeight > cardHeightForMaxWidth) {

      // عدل علي طول الكارت اول بأول

      memoryCards.style.height = (cardHeightForMaxWidth \* rows) + "px";

      memoryCards.style.width = document.getElementById('memory--app-container').offsetWidth + "px";

      memoryCards.style.top = ((cardMaxHeight \* rows - (cardHeightForMaxWidth \* rows)) / 2) + "px";

    }

    else {

      // عدل علي عرض الكارت اول باول

      memoryCards.style.width = (cardWidthForMaxHeight \* columns) + "px";

      memoryCards.style.height = document.getElementById('memory--app-container').offsetHeight + "px";

      memoryCards.style.top = 0;

    }

  };

**لما حجم الشاشة يتغير اعمل الفاكشن اللي فوق**

window.addEventListener('resize', function() {

    buildLayout($.cards, $.settings.rows, $.settings.columns);

  }, true);

**الفاكشن اللي بتبني كارت واحد في الhtml مش محتاجة شرح هي بتبني elements بترتيب**

var buildCardNode = function (index, card, width, height) {

    var flipContainer = document.createElement("li");

    var flipper = document.createElement("div");

    var front = document.createElement("a");

    var back = document.createElement("a");

    flipContainer.index = index;

    flipContainer.style.width = width;

    flipContainer.style.height = height;

    flipContainer.classList.add("flip-container");

    if (card.isRevealed) {

      flipContainer.classList.add("clicked");

    }

    flipper.classList.add("flipper");

    front.classList.add("front");

    front.setAttribute("href", "#");

    back.classList.add("back");

    back.classList.add("card-" + card.value);

    if (card.isMatchingCard) {

      back.classList.add("matching");

    }

    back.setAttribute("href", "#");

    flipper.appendChild(front);

    flipper.appendChild(back);

    flipContainer.appendChild(flipper);

    flipContainer.addEventListener('click', handleFlipCard);

    return flipContainer;

  };