

Pràctica P002 JOC MULTIMÈDIA MEMÒRIA.

Objectius.

En aquesta segona pràctica desenvoluparem un joc de memòria. Altre cop, la interfície gràfica d'usuari es realitzarà amb HTML5 i CSS3 utilitzant les transicions i transformacions. La interacció de l'usuari i l'accés a la IGU per part del programa serà a través de la llibreria jQuery.

Treball a realitzar en grups de 2 alumnes

Descripció general:

En aquesta activitat es consoliden els coneixements sobre l'entorn de desenvolupament web que s'usarà en l'assignatura. L'alumne es familiaritzarà en l'edició del programa així com la prova i depuració del mateix. S'utilitzaran les eines de desenvolupament consola (per veure els errors de programació), depurador (per executar pas a pas el programa) i l'inspector de variables (per veure la traça del programa).

Caldrà completar la codificació de part de la lògica d'un joc la interfície del qual seran diferents elements HTML. L'estudiant disposarà d'un preprojecte (inacabat) del joc que ha d'implementar.

Material de suport:

Enunciat de la pràctica.

Preprojecte del joc on ja està elaborada la interfície gràfica d'usuari amb objectes del DOM.

Lliuraments i vincles amb l'avaluació:

Cal entregar el projecte acabat, és a dir, que compleixi amb totes les especificacions de l'enunciat.

El resultat de l'activitat s'incorporarà a l'avaluació de l'assignatura amb un valor del 20% de la nota final.

Objectius específics:

En finalitzar l'activitat l'estudiant o estudianta ha de ser capaç de:

Conèixer i usar adequadament l'entorn de desenvolupament: escriure, provar i depurar el programa.

Programar l'animació d'objectes de la interfície d'usuari.

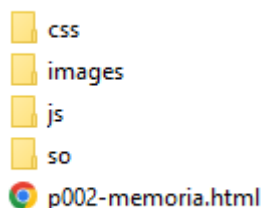
Accedir als elements HTML de la interfície d'usuari per a la seva modificació.

Incorporar so a l'execució del joc

Infraestructura i organització .

Acompanyant aquest enunciat se us lliura el material necessari inicial com a punt de partida, amb la següent estructura:

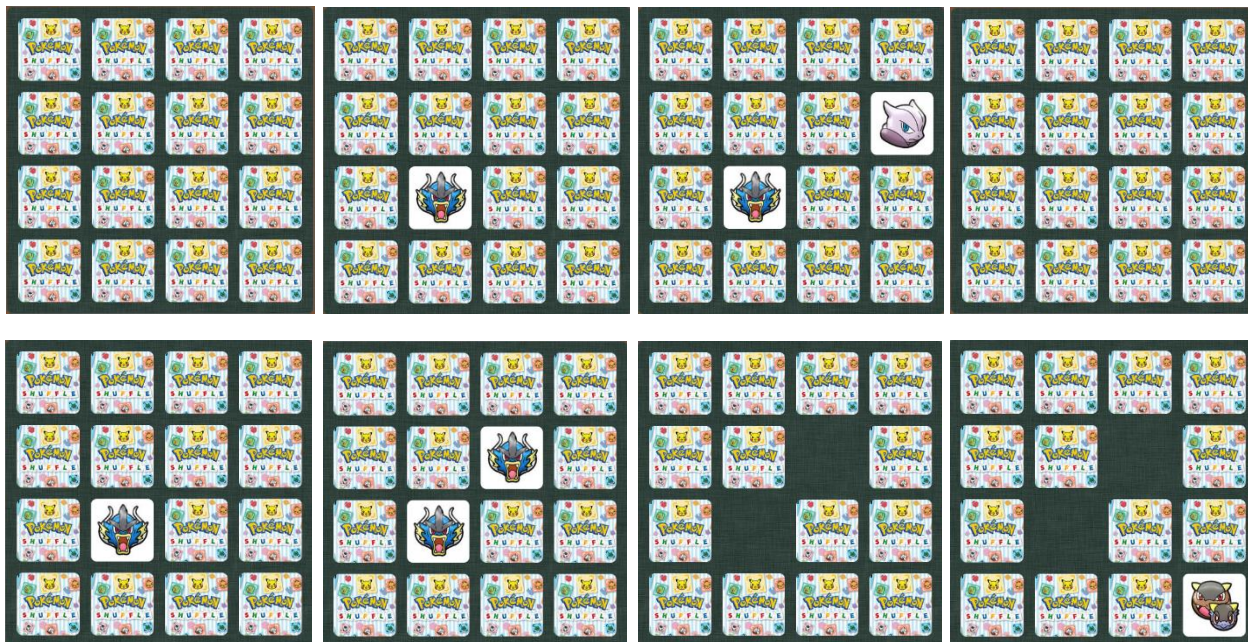
Nombre



Requeriments del projecte

La interfície gràfica d'usuari consta d'un joc de 16 cartes aparellades de dos en dos i posades boca avall. L'usuari al fer clic en una carta qualsevol, aquesta dóna la volta per mostrar el seu valor. Quan hi ha dues cartes girades, mostrant el seu valor, el programa comprova si són iguals o no. En el cas de que siguin iguals, desapareixen del taulell i si no són iguals, es tornen a girar per amagar el seu valor. El joc consisteix en trobar les parelles amb el mínim nombre de clics.

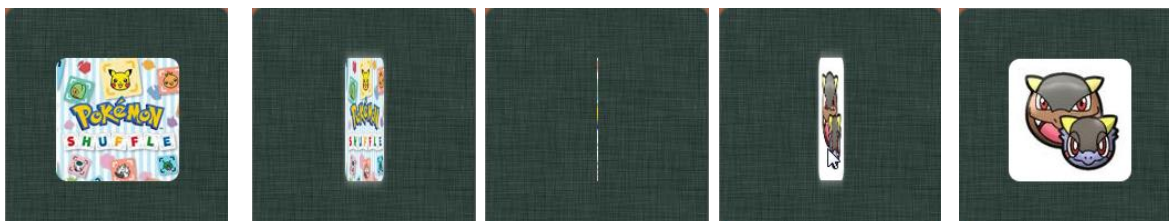
Observeu els primers clics en una partida qualsevol:



Podeu jugar amb qualsevol tipus d'imatges, no necessàriament de pokemon. Però, cal que estiguin totes les imatges agrupades en una de sola, és a dir, ha de ser un "sprite":



Cal que la carta giri aplicant una transformació de tipus gir 3D en l'eix y de la carta i una transició de 0,5 segons en resposta al clic de l'usuari.



Això s'aconsegueix amb el següent HTML+CSS+JS:

HTML:

```
<div class="carta">
  <div class="cara davant"></div>
  <div class="cara darrera"></div>
</div>
```

CSS:

```
.carta {
width: 111px; height: 111px; /* mides de la carta */
position: absolute; /* per poder moure la posició: "top" i "left" de la carta*/
top: 45px; left: 45px;
}

.cara {
border-radius: 10px;
width: 100%; height: 100%;
position: absolute; /* per poder superposar les dues cares */

transition-property: transform;
transition-duration: 5s;

backface-visibility: hidden; /* per no veure el background quan està girat més de 90° */
}

.darrera {
background: #999 url(images/pokemon.jpg) -777px 0px;
}

.davant {
background: #eee url(images/pokemon.jpg);
transform: rotate3d(0,1,0,-180deg); /* girem pel eix y, angle=-180° */
/* no es veurà degut a backface-visibility: hidden;*/
}

.cart-a-girada .darrera {
transform: rotate3d(0,1,0,180deg); /* girem pel eix y, angle=180°, venim de 0° */
}

.cart-a-girada .davant {
transform: rotate3d(0,1,0,0deg); /* girem pel eix y, angle=0°, venim de -180°*/
}
```

Javascript:

```
$(".carta").click(function() {
// afegim/traiem la classe "carta-girada" en clics successius.
// el navegador animarà la transició dels estils entre l'estat actual i l'estat de carta girada.
$(this).toggleClass("carta-girada");
});
```

En el preprojecte podeu trobar una aplicació amb 2 x 2 cartes que podem girar tantes vegades com vulguem. Utilitzeu el codi d'aquesta aplicació com a punt de partida per a la pràctica. Reproduïm el codi tot seguit:

```
<!DOCTYPE html>
<html lang="cat">
<head>
  <meta charset="utf-8">
  <title>2x2 cartes, totes iguals</title>
  <style>
  body {
    text-align: center;
    background: #777;
    color: #ddd;
  }

  #tauler {
    border-radius: 10px;
    border: 1px solid #666;
    position: relative;
    background: #232 url(images/table.jpg);
    margin: 0 auto;
    box-shadow: 0 -1px 0 #999, 0 1px 0 #999, inset 0;
  }

  .carta {
    width: 111px; height: 111px; /* pokemon */
    position: absolute;
  }

  .carta:hover .cara {
    box-shadow: 0 0 10px #aaa;
  }

  .cara {
    border-radius: 10px;
    width: 100%; height: 100%;
    position: absolute; /* per poder moure la carta */

    transition-property: transform;
    transition-duration: .5s;

    backface-visibility: hidden;
  }

  .darrera {
    background: #999 url(images/pokemon.jpg) -777px 0px;
  }

  .davant {
    background: #efefef url(images/pokemon.jpg);
    transform: rotate3d(0,1,0,-180deg);
  }

  .carta-girada .darrera {
    transform: rotate3d(0,1,0,180deg);
  }

  .carta-girada .davant {
    transform: rotate3d(0,1,0,0deg);
  }

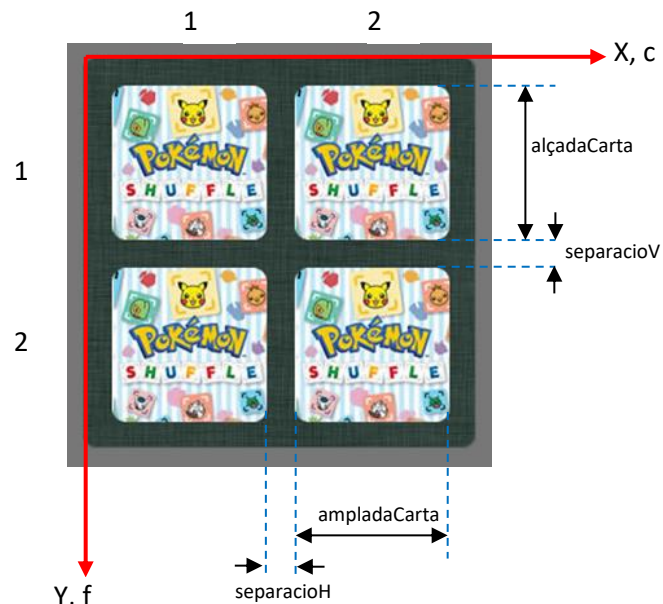
  /* pokemon */
  .carta1{background-position: -111px 0; }

</style>
</head>

<body>
<header>
  <h1>n cartes, totes iguals</h1>
</header>

<div id="tauler">
  <div class="carta" id="f1c1"> <div class="cara darrera"></div><div class="cara davant"></div> </div>
  <div class="carta" id="f1c2"> <div class="cara darrera"></div><div class="cara davant"></div> </div>
  <div class="carta" id="f2c1"> <div class="cara darrera"></div><div class="cara davant"></div> </div>
  <div class="carta" id="f2c2"> <div class="cara darrera"></div><div class="cara davant"></div> </div>
</div>

<footer>
  <p>Fes clic en una carta per donar-li la volta</p>
</footer>
```



```
<script src="javascript/jquery-3.1.1.min.js"></script>
<script>
var ampladaCarta, alçadaCarta;
var separacioH=20, separacioV=20;
var nFiles=2, nColumnes=2;

var jocCartes = [
  'carta1', 'carta1',
  'carta1', 'carta1'
];

function barrejarCartes(){
}

$(document).ready(function(){
  var f, c, carta;

  // barregem les cartes
  barrejarCartes();

  // mirem les mides de la carta
  ampladaCarta=$("#carta").width(); alçadaCarta=$("#carta").height();

  // mida del tauler
  $("#tauler").css({
    "width": "280px",
    "height": "280px"
  });

  // inicialitzem totes les cartes: posició i figura
  for(f=1; f<=nFiles; f++){
    for(c=1; c<=nColumnes; c++){
      carta=$("#f"+f+"c"+c);
      // posició
      carta.css({
        "left": ((c-1)*(ampladaCarta+separacioH)+separacioH)+"px",
        "top": ((f-1)*(alçadaCarta+separacioV) +separacioV)+"px"
      });
      // figura
    }
  }

  $("#carta").click(function(){
    $(this).toggleClass("carta-girada");
  });
});
</script>
</body>
</html>
```

Objectiu del joc.

Observeu que en la pràctica el que cal fer és gestionar l'event clic de tal manera que:

1. es seleccionin dues cartes amb clics consecutius
2. es mira si les dues cartes són iguals (la mateixa figura):
 - a. si són iguals, s'eliminen o s'oculten
 - b. si no són iguals, es tornen a girar.

Tasques a realitzar.

Se us dona el codi javascript bàsic de l'arxiu p002-memoria.js que heu de completar i ampliar.

1. El codi ha de permetre generar jocs mínim des de 2*2 fins a 4*4 cartes, nFiles=4 i nColumnes=4
 - a. Mínim ha de permetre modificar el valor de les variables (nFiles i nColumnes) per codi i el joc s'ha de generar i adaptar en funció d'aquests paràmetres
 - b. Addicional si és possible que sigui l'usuari el que decideixi el nombre de cartes amb les que jugar, o hi ha diferents nivells de joc que provoquen el canvi en el número de cartes
2. Les cartes han d'estar barrejades amb algun tipus de funció random. No poden aparèixer de manera consecutiva.
3. Els contenidors de les cartes (div) s'han de generar en temps d'execució, actualitzant el contingut HTML del contenidor "TAULER"
4. Les mides del "TAULER" s'han d'ajustar, en temps d'execució, a les mides de les cartes triades per jugar i a la separació (marge) entre elles i les vores del tauler, i al número de files i/o columnes definides.
5. Detecteu la situació de parella de cartes girades: si són de la mateixa figura cal eliminar-les, si són de figures diferents cal tornar-les a girar.
6. Detecteu la situació de final de partida: totes les cartes eliminades. Feu aparèixer un missatge quan es doni aquesta situació i permeteu tornar a jugar una altra partida.
7. Detecteu si l'usuari ha fet tants clics com el triple de cartes del joc. En aquest cas, l'usuari perd la partida. Feu aparèixer un missatge quan es doni aquesta situació i permeteu tornar a jugar una altra partida

Si heu arribat fins aquí, ja teniu la pràctica aprovada. Ara podeu fer alguna d'aquestes ampliacions per tal d'obtenir la màxima nota:

Possibles ampliacions del projecte.

1. Que hi hagi un marcador, és a dir, a la interfície de l'usuari s'indica el nombre de parelles de cartes girades que s'ha realitzat fins el moment i la quantitat de clics i/o la quantitat de clics restants per perdre la partida.
2. Que hi hagi una limitació de temps per resoldre el joc.
3. Que hi hagi una etapa inicial de configuració del joc (setup) amb algunes de les següents opcions:
 - a. Que es pugui jugar amb un nombre determinat de cartes, és a dir, l'usuari entra el nombre de cartes amb les que vol jugar.
 - b. Que es pugui triar amb quin joc de cartes jugar.
 - c. Que les cartes es col·loquin al tauler amb una animació des d'una pila de cartes.
4. Afegir efectes sonors: clic, gir de la carta, desaparició de la carta, final de joc, etc.

Lliuraments.

Dins del termini establert per al lliurament de la pràctica en el grup corresponent, cal:

1. Una carpeta zip amb tot el contingut necessari per a que la web sigui 100% funcional amb el nom P002_idgrup.
2. Es valora els 7 punts definits com a tasques a realitzar, més les ampliacions addicionals per aconseguir una nota superior a l'aprobat. També l'ampliació i personalització del codi CSS i l'aplicació d'efectes visuals (animacions) i/o sons. En aquest últim cas, cal documentar aquestes aplicacions, acompanyant l'entrega amb un document o afegint un comentari destacat al codi de l'aplicació