
# Multimédia nappali gyakorlat

## Tizenötös játék

A tananyaghoz készült videó az alábbi linken érhető el: https://www.youtube.com/...

**Utoljára módosítva**
2020. április 29.

**Észrevételek, javaslatok**
✉ mkatona@inf.u-szeged.hu

# Tizenötös játék (jQuery)


Megoldás: megoldas.zip

# Megoldás menete

Indítsuk el a PhpStorm-ot vagy Webstorm-ot és hozzunk létre egy új projektet, amiben legyen egy `index.html` is. A feladat megoldáshoz nincs szükség külső képállomány(ok)ra. Hozzunk létre egy DIV-et, mely a játékteret fogja magában foglalni. A feladatmegoldáshoz használni fogjuk a jQuery függvénykönyvtárat, ezért szükséges annak a hozzáadása a weboldalhoz. Amikor sikerül megfelelő sorendbe rakni a számokat, akkor megjelenik a "Gratulálunk, nyertél." felirat. Ezt már most elhelyezzük a játéktéren, de majd látni fogjuk, hogy a megjelenítését nem engedélyezzük. Egy gomb is helyet foglal majd a számok alatt, mert bármikor lehetőség lesz újrakezdeni a játékot a megnyomásával.

```
<!DOCTYPE html>
<html lang="hu">
<head>
<meta charset="UTF-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<title>Tizenotos jatek</title>
</head>
<body>
  <div id="gameArea"></div>
  <p id="win">Gratulalok, nyertel.</p>
  <button id="newGame">Uj jatek</button>
</body>
</html>
```

Mielőtt a tényleges megvalósítást megtekintenénk, nézzük meg a különböző stílusformázási beállításokat. A csempéknek, amiket mozgatni fogunk, beállítunk háttérszínt arra az állapotra, amikor felé megyünk, illetve alapértelmezetten. Az itt látható beállítások az élvezhető vizuális megjelenést teszik lehetővé.

```
<style>
  body {
    margin: 0;
    padding: 20px;
```

```
  }

  .tile {
    position: absolute;
    border-width: 1px;
    border-style: solid;
    background: #ffe99e;
    border-color: black;
    text-align: center;
    color: black;
    cursor: pointer;
  }

  .tile:hover {
    background-color: #ffa12a;
  }

  #gameArea {
    position: relative;
    padding: 20px;
    background-color: #1c5980;
    border: solid black 3px;
  }
</style>
```

Tekintsük meg a deklarált, definiált változók szerepét. Az `N` értékadással befolyásolhatjuk, hogy hány sora, oszlopa legyen a játéknak. A `baseSize` egy csempe méretét definiálja. A játéktér belső margójának szélességét az `innerPadding` értékével állíthatjuk be. Eltároljuk a játékteret a `gameArea` változóban. Az `emptyTile` az üres mezőt tárolja, míg az `allBorderWidth` is vizuális beállítás a csempe szélességét növelő szegélyre.

```
var N = 4;
var baseSize = 100;
var innerPadding = 20;
var gameArea = $('#gameArea');
var emptyTile;
var allBorderWidth = 4;
```

A játéktér alapterülete az általunk megadott `baseSize` értéke. Ez beszorzódik az elemek (`N`) számával a tér szélességét és magasságát illetően is. Az "Új játék" feliratot tartalmazó gombhoz eseményfigyelést rendelünk és amennyiben azt lenyomjuk, meghívjuk a `newGame` függvényt. Alapvetően a játék inicializálásához is meghívjuk.

```
gameArea.css({
  width: baseSize*N+"px",
  height: baseSize*N+"px"
});

$("#newGame").on('click', newGame);

newGame();
```

Ez a függvény minden fontos dolgot magában foglal, ami a játszhatósághoz és a megfelelő működéshez, pálya felépítéséhez szükséges. Részletesen vizsgáljuk az egyes elemeit a továbbiakban.

```
function newGame(){
  $('#win').hide();

  var order = [];
  for (var i=0; i<N*N-1; i++){
    order.push(i+1);
  }
  shuffle(order);

  emptyTile = {x: N-1, y: N-1};

  gameArea.html("");

  for (var i=0; i<N*N-1; i++){

    var indexY = Math.floor(i/N);
    var indexX = i%N;

```

```
    var tile = $('<div class="tile"></div>');
    tile.attr('iX', indexX);
    tile.attr('iY', indexY);
    tile.attr('tileNumber', order[i]);

    gameArea.append(tile);

    tile.css({
      top: innerPadding+indexY*baseSize+"px",
      left: innerPadding+indexX*baseSize+"px",
      width: baseSize-allBorderWidth+"px",
      height: baseSize-allBorderWidth+"px",
      'line-height': baseSize-allBorderWidth+"px",
      'font-size': baseSize/2+"px"
    }).text(order[i]);

    tile.on('click', tileClick);
  }
}
```

A nyertél feliratot rejtő DIV-et rejtsük el, hisz a játék elején ez még nem áll fenn. Ezt követően töltsünk fel egy tömböt annyi értékkel, amekkora a pályánk. Tehát, jelen esetben 4-ben adtuk meg a a sorok és oszlopok számát, ami azt jelenti, hogy N*N-1 darab elem lesz, mivel van egy üres mező is. A számozást 1-től Kezdjük. Annak érdekében, hogy ne sorban legyenek a számok, megkeverjük ezt a tömböt.

```
$('#win').hide();

var order = [];
for (var i=0; i<N*N-1; i++){
  order.push(i+1);
}
shuffle(order);
```

A megkeverés azt fogja jelenteni, hogy végigmegyünk a tömbön, generálunk egy random számot és az aktuális pozíciójához hozzáadjuk azt, majd elhelyezzük az új indexre cserével. Figyeljünk arra, hogy az `index` értéke 0 is lehet, mivel van elem a 0. tömbindexen.

```
function shuffle(inputArray){
  for (var i=0; i<N*N-1; i++){

    var index = i + Math.floor(Math.random()*(inputArray.length-i));

    var tmp = inputArray[i];
    inputArray[i] = inputArray[index];
    inputArray[index] = tmp;
  }
  return inputArray;
}
```

Megadjuk az üres mező indexeit, ami a jobb alsó sarok lesz. A játékterületet kiürítjük. Erre korábban a `remove()` függvényt láhattunk. Ezt követően feltöltjük a teret. Meghatározzuk, hogy mik lesznek a sor- és oszlopindexek, majd hozzáadunk egy új csempét. Beállítjuk attribútimként az aktuális pozíciót és kiveszünk egy számot, melyet majd megjelenítünk a csempén a korábban összekevert listából. Hozzáadjuk a játéktérhez és ténylegesen is a megfelelő koordinátára rajzoljuk ki a csempét és a számot. Kattintásra élesedő eseményt rendelünk a csempékhez a mozgatás kedvéért.

```
emptyTile = {x: N-1, y: N-1};

gameArea.html("");

for (var i=0; i<N*N-1; i++){

  var indexY = Math.floor(i/N);
  var indexX = i%N;

  var tile = $('<div class="tile"></div>');
  tile.attr('iX', indexX);
  tile.attr('iY', indexY);
  tile.attr('tileNumber', order[i]);

  gameArea.append(tile);

```

```
  tile.css({
    top: innerPadding+indexY*baseSize+"px",
    left: innerPadding+indexX*baseSize+"px",
    width: baseSize-allBorderWidth+"px",
    height: baseSize-allBorderWidth+"px",
    'line-height': baseSize-allBorderWidth+"px",
    'font-size': baseSize/2+"px"
  }).text(order[i]);

  tile.on('click', tileClick);
}
```

Amennyiben bármelyik csempén kattintunk, úgy lekérjük annak az elemnek az x és y indexeit, melyeket attribútumként adtunk hozzá korábban. Ezt követően megvizsgáljuk, hogy a kattintott elem felcserélhető-e az üres mezővel, mert ha az nincs a 4 szomszédságában, akkor nem. Amennyiben igen, tehát a `canSwap` értéke 1, akkor a csempére meghívjuk az animációt, mely az új helyére mozgatja a csempét. Közben frissítjük a csempe pozíciójának attribútumait és az üres csempe is felveszi a kattintott pozíciót. A cserét követően megvizsgáljuk, hogy így már helyes sorrendben vannak-e az elemek, meghívjuk a `testWin()` függvényt. Ha igen, akkor levesszük az eseményfigyelőt a csempékről és megjelenítjül a korábban elrejtett "Gratulálunk, nyertél." feliratot.

```
function tileClick(){

  var iX = $(this).attr('iX');
  var iY = $(this).attr('iY');

  var canSwap = Math.abs(iX - emptyTile.x) + Math.abs(iY - emptyTile.y) == 1;
  if (canSwap){
    $(this).animate({
    top: innerPadding + emptyTile.y*baseSize+"px",
    left: innerPadding + emptyTile.x*baseSize+"px"
    });

    $(this).attr('iX', emptyTile.x);
    $(this).attr('iY', emptyTile.y);
    emptyTile.x = iX;
    emptyTile.y = iY;
  }
  var win = testWin();
  if (win){
    gameArea.find('.tile').off('click');
    $('#win').show();
  }
}
```

A sorrend tesztelséért a `testWin()` függvény felel. Először leellenőrizzük, hogy az üres mező a tábla végén található-e. Ha nem, akkor már egyből teljesítetlen a kirakás. Alapvetően azt feltételezzük, hogy kiraktuk, ezért az `allInOrder` értékének `true`-t adunk. Ezután végigmegyünk az összes csempén és megvizsgáljuk azok `tileNumber` attribútumát, mely az adott csempén található számot tartalmazza. Megnézzük, hogy sorban vannak-e a számok és ha már valahol nem, akkor megváltoztatjuk az `allInOrder` értékét `false`-ra. Ezzel az értékkel fogunk visszatérni.

```
function testWin(){

  if (emptyTile.x != N-1 || emptyTile.y != N-1) {
    return false;
  }
  var allInOrder = true;
  var lastIndex = -1;

  gameArea.find('.tile').each(function(){
    var tileNumber = parseInt($(this).attr('tileNumber'));
    if (lastIndex > tileNumber) {
      allInOrder = false;
    }
    lastIndex = tileNumber;
  });
  return allInOrder;
}
```