Créez un jeu de Sokoban en Flash
Code intégral

```
level1 = new Array();
level1[0] =[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level1[1] =[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level1[2] =[0,0,0,0,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0];
level1[3] =[0,0,0,0,2,1,1,1,2,0,0,0,0,0,0,0,0,0,0,0];
level1[4] =[0,0,0,0,2,1,1,1,2,0,0,0,0,0,0,0,0,0,0,0];
level1[5] =[0,0,2,2,2,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0];
level1[6] =[0,0,2,1,1,1,1,1,1,2,0,0,0,0,0,0,0,0,0,0];
level1[7] =[2,2,2,1,2,1,2,2,1,2,0,0,0,2,2,2,2,2,2,0];
level1[8] =[2,1,1,1,2,1,2,2,1,2,2,2,2,2,1,1,3,3,2,0];
level1[9] =[2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,2,0];
level1[10]=[2,2,2,2,2,1,2,2,2,1,2,1,2,2,1,1,3,3,2,0];
level1[11]=[0,0,0,0,2,1,1,1,1,1,2,2,2,2,2,2,2,2,2,0];
level1[12]=[0,0,0,0,2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0];
level1[13]=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level1[14]=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level1[15]=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
poussoir1 = [11,10];
boules1 = [[5,4],[5,6],[7,5],[7,6],[2,9],[5,9]];
level2 = new Array();
level2[0] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level2[1] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level2[2] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level2[3] = [0,0,0,2,2,2,2,2,2,2,2,2,2,2,0,0,0,0,0,0];
level2[4] = [0,0,0,2,3,3,1,1,2,1,1,1,1,1,2,2,2,0,0,0];
level2[5] = [0,0,0,2,3,3,1,1,2,1,1,1,1,1,1,1,2,0,0,0];
level2[6] = [0,0,0,2,3,3,1,1,2,1,2,2,2,2,1,1,2,0,0,0];
level2[7] = [0,0,0,2,3,3,1,1,1,1,1,1,2,2,1,1,2,0,0,0];
level2[8] = [0,0,0,2,3,3,1,1,2,1,2,1,1,1,1,2,2,0,0,0];
level2[9] = [0,0,0,2,2,2,2,2,2,1,2,2,1,1,1,1,2,0,0,0];
level2[10]= [0,0,0,0,0,2,1,1,1,1,1,1,1,1,1,1,2,0,0,0];
level2[11]= [0,0,0,0,0,2,1,1,1,1,2,1,1,1,1,1,2,0,0,0];
level2[12]= [0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2,0,0,0];
level2[13]= [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level2[14]= [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level2[15]= [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
poussoir2 = [10,07];
boules2 = [[7,10],[9,6],[10,5],[13,5],[13,8],[14,9],[14,10],[12,9],[12,10],[10,10]];
level3 = new Array();
level3[0] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level3[1] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level3[2] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level3[3] = [0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,0,0,0];
level3[4] = [0,0,0,0,0,0,0,0,0,2,1,1,1,1,1,1,2,0,0,0];
level3[5] = [0,0,0,0,0,0,0,0,0,2,1,1,2,1,1,2,2,0,0,0];
level3[6] = [0,0,0,0,0,0,0,0,0,2,1,1,1,1,1,2,0,0,0,0];
level3[7] = [0,0,0,0,0,0,0,0,0,2,2,1,1,1,1,2,0,0,0,0];
level3[8] = [0,2,2,2,2,2,2,2,2,2,1,1,1,2,1,2,2,2,0,0];
level3[9] = [0,2,3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,2,0,0];
level3[10] = [0,2,2,2,3,3,3,1,1,1,1,1,1,1,1,1,1,2,0,0];
level3[11] = [0,2,3,3,3,3,3,1,1,2,2,2,2,2,2,2,2,2,0,0];
level3[12] = [0,2,2,2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0];
level3[13] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level3[14] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
level3[15] = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
poussoir3 = [15,04];
boules3 = [[10,10],[11,5],[11,6],[11,7],[11,8],[11,9],[13,5],[14,6],[13,7],[14,9],[13,10]];
function interpretation() {
   historique = new Array();
   suivante = new Object();
   postsuiv = new Object();
   for(y = 0; y < 16; y++) {
      if(y < 10) { y = "0" + y; }
```

```
      for(x = 0; x < 20; x++) {
         if(x < 10) { x = "0" + x; }
        _root["case"+y+x].attachMovie("aspect"+_root["level" +
noniveau][Number(y)][Number(x)], this._name, 0);
      }
   }
   poussoir._x = 32 * _root["poussoir" + noniveau][0];
   poussoir._y = 32 * _root["poussoir" + noniveau][1];
   for(i = 0; i < _root["boules" + noniveau].length; i++) {
      boule.duplicateMovieClip("boule" + i, this.getNextHighestDepth());
      _root["boule" + i]._x = 32 * _root["boules" + noniveau][i][0];
      _root["boule" + i]._y = 32 * _root["boules" + noniveau][i][1];
   }
}
function bougerpoussoir(codetouche:Number) {
   switch(codetouche) {
      case Key.RIGHT:
      poussoir._x += 32; historique.push("d"); break;
      case Key.LEFT:
      poussoir._x -= 32; historique.push("g"); break;
      case Key.DOWN:
      poussoir._y += 32; historique.push("b"); break;
      case Key.UP:
      poussoir._y -= 32; historique.push("h"); break;
   }
}
function bougerboule(codetouche:Number, noboule:Number) {
   switch(codetouche) {
      case Key.RIGHT:
      _root["boule" + noboule]._x += 32; historique.push("d" + noboule); break;
      case Key.LEFT:
      _root["boule" + noboule]._x -= 32; historique.push("g" + noboule); break;
      case Key.DOWN:
      _root["boule" + noboule]._y += 32; historique.push("b" + noboule); break;
      case Key.UP:
      _root["boule" + noboule]._y -= 32; historique.push("h" + noboule); break;
   }
}
function comptage() {
   points = 0;
   for(y = 0; y < 16; y++) {
      if(y < 10) { y = "0" + y; }
      for(x = 0; x < 20; x++) {
         if(x < 10) { x = "0" + x; }
         if(level1[Number(y)][Number(x)] == 3) {
            for(i = 0; i < boules1.length; i++) {
               if(_root["boule" + i]._x / 32 == x && _root["boule" + i]._y / 32 == y) {
                  points++;
                  if(points == _root["boules" + noniveau].length) {
                     noniveau++;
                     interpretation();
                  }
               }
            }
         }
      }
   }
}
ecouteur = new Object();
ecouteur.onKeyDown = function() {
   posx = poussoir._x / 32;
   posy = poussoir._y / 32;
   if(Key.getCode() == Key.RIGHT) {
      suivante.posx = posx + 1;
      suivante.posy = posy;
      postsuiv.posx = posx + 2;
      postsuiv.posy = posy;
   }
```

```
   else if(Key.getCode() == Key.LEFT) {
      suivante.posx = posx - 1;
      suivante.posy = posy;
      postsuiv.posx = posx - 2;
      postsuiv.posy = posy;
   }
   else if(Key.getCode() == Key.DOWN) {
      suivante.posx = posx;
      suivante.posy = posy + 1;
      postsuiv.posx = posx;
      postsuiv.posy = posy + 2;

   }
   else if(Key.getCode() == Key.UP) {
      suivante.posx = posx;
      suivante.posy = posy - 1;
      postsuiv.posx = posx;
      postsuiv.posy = posy - 2;
   }
   suivante.aspect = _root["level" + noniveau][suivante.posy][suivante.posx];
   postsuiv.aspect = _root["level" + noniveau][postsuiv.posy][postsuiv.posx];
   for(i = 0; i < _root["boules" + noniveau].length; i++) {
      if(_root["boule" + i]._x / 32 == suivante.posx) {
         if(_root["boule" + i]._y / 32 == suivante.posy) {
                        suivante.boule = i; break;
         }
      }
      else { suivante.boule = undefined; }
   }
   for(i = 0; i < _root["boules" + noniveau].length; i++) {
      if(_root["boule" + i]._x / 32 == postsuiv.posx) {
         if(_root["boule" + i]._y / 32 == postsuiv.posy) {
            postsuiv.boule = i; break;
         }
      }
      else { postsuiv.boule = undefined; }
   }
   if(suivante.aspect == 1 || suivante.aspect == 3) {
      if(suivante.boule == undefined) {
         bougerpoussoir(Key.getCode());
      }
      else if(suivante.boule != undefined) {
         if(postsuiv.aspect == 1 || postsuiv.aspect == 3) {
            if(postsuiv.boule == undefined) {
               bougerpoussoir(Key.getCode());
               bougerboule(Key.getCode(), suivante.boule);
            }
         }
      }
   }
   comptage();
}
Key.addListener(ecouteur);
undo.onRelease = function() {
   if(historique[historique.length - 1].length == 1) {
      switch(historique[historique.length - 1]) {
         case "d":
         poussoir._x -= 32; break;
         case "g":
         poussoir._x += 32; break;
         case "h":
         poussoir._y += 32; break;
         case "b":
         poussoir._y -= 32; break;
      }
   }
   else if(historique[historique.length - 1].length == 2) {
      noboule = historique[historique.length - 1].charAt(1);
```

```
      switch(historique[historique.length - 1].charAt(0)) {
         case "d":
         poussoir._x -= 32;
         _root["boule" + noboule]._x -= 32; break;
         case "g":
         poussoir._x += 32;
         _root["boule" + noboule]._x += 32; break;
         case "h":
         poussoir._y += 32;
         _root["boule" + noboule]._y += 32; break;
         case "b":
         poussoir._y -= 32;
         _root["boule" + noboule]._y -= 32; break;
      }
      historique.pop();
   }
   historique.pop();
}
noniveau = 1;
interpretation();
```