



# PROJET AJAX RETENTION

## Résumé

Ce projet a été réalisé seul dans le cadre du cours d'Ajax en 2ème année.

Adrien MOUSTY  
moustyadrien@hotmail.com

## Table des matières

GUI.....	2
Description .....	2
Exemples .....	2
Langages.....	3
JavaScript.....	3
RCP.....	5
HTML - CSS.....	6
Résumé.....	7

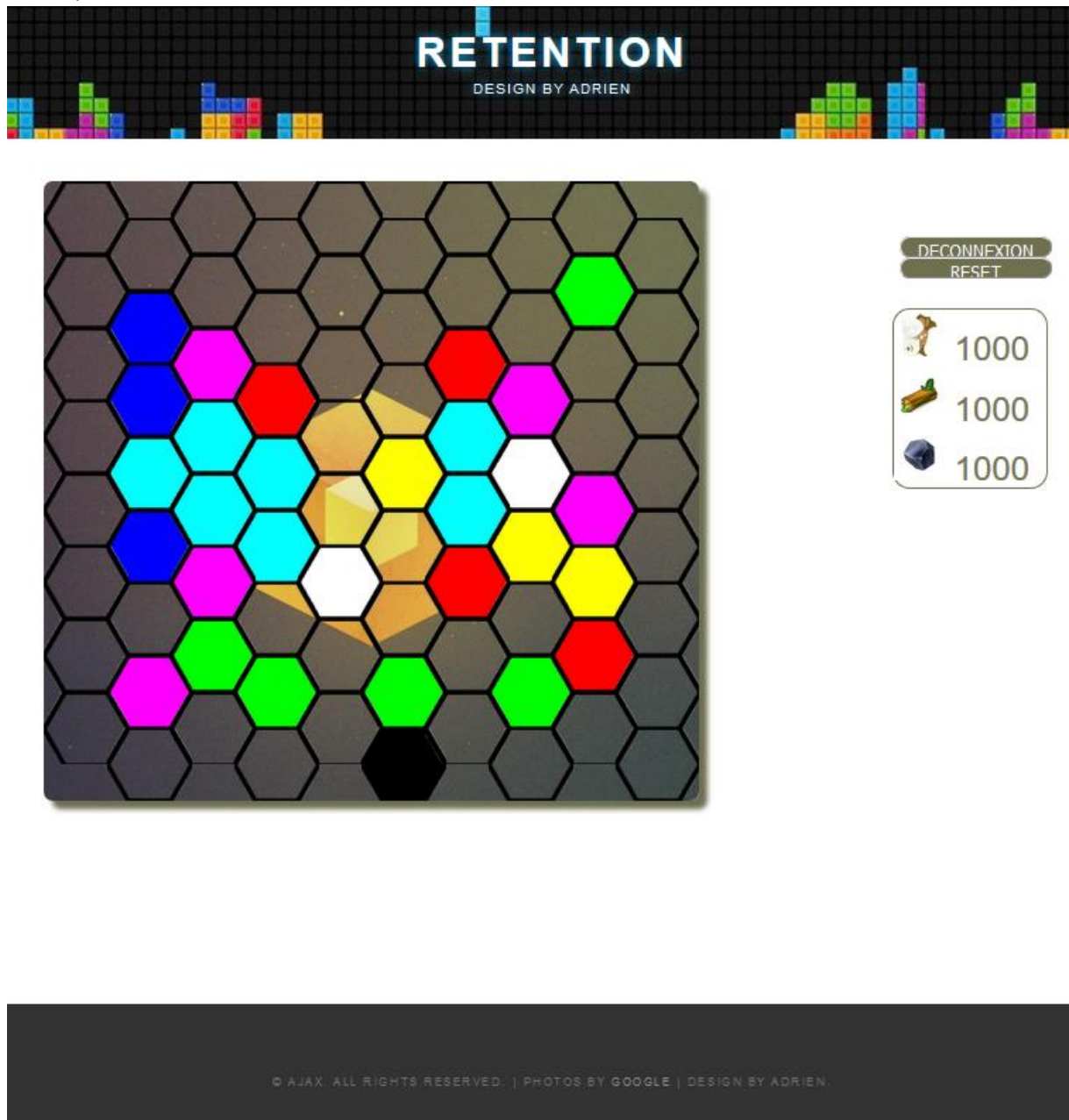
## GUI

### Description

Le but est de développer un écran de connexion/inscription, un écran pour afficher la map et un pour afficher le village de la personne.

C'est un élément qui n'a pas vraiment été mis en avant dans ce projet, le principal élément ici était de faire un code propre.

### Exemples



## Langages

### JavaScript

Fonction d'affichage de la map. Appel à une autre fonction pour charger les données + phénomène de closure.

```
354 /* ***** */
355 /* FONCTIONS D’AFFICHAGE - MASQUAGE */
356 /* ***** */
357
358 // ////////////////////////////////////
359 // AffichageMap() : Affiche une certaine couleur selon les éléments envoyés ////////////////////////////////////.
360 // Entrées: Les couleurs de la map afin de l'envoyer à ChangerCouleur()
361 // ////////////////////////////////////
362 function AffichageMap(str)
363 {
364     var hexagone;
365     var top = 0;
366     var left = 0;
367
368     $('carte').innerHTML = ""; // On vide la map.
369     var recup = str.split("|");
370     // -> séparer une première fois les groupes entre pipe (|) en x éléments
371     for(var k = 0; k < recup.length; k++)
372     {
373         recup[k] = recup[k].split(",");
374         //-> on choppe un tableau double dimension recup[i][j] car on a sépare à nouveau en y éléments
375     }
376     for (var i = 0; i < x; i++)
377     {
378         for(var j = 0; j < y; j++)
379         {
380             posVillage = 'h';
381             posi = i + "";
382             posj = j + "";
383             posVillage+= i;
384             posVillage+= j;
385             hexagone = document.createElement("div");
386             hexagone.id=posVillage;
387             ChangerCouleur(hexagone, recup[i][j])
388
389             /* PLACEMENT DES ELEMENTS */
390             top = i*86+(j%2)*43;
391             left = j*75;
392             hexagone.style.top = top + 'px';
393             hexagone.style.left = left + 'px';
394             hexagone.onclick = VueVillageGCB(hexagone, i, j);
395             $('carte').appendChild(hexagone);
396             /* -> on ajoute l'element qu'on a crée à la fin d'un élément, ici map.*/
397         }
398     }
399     /* pour gérer les coordonnées impaires*/
400     $('carte').style.width = (y*73 + 43) + 'px';
401     $('carte').style.height = (x*86 + 43) + 'px';
402 }
403
```

```

205  /***** FONCTIONS GEN CALL BACK *****/
206  /*
207  *****/
208
209  ///////////////////////////////////////////////////////////////////
210  // VueVillageGCB() : Permet d'éviter le phénomène de closure, permet en outre de générer
211  /// Entrées: L'id du village ainsi que sa position x et y
212  ///////////////////////////////////////////////////////////////////
213  function VueVillageGCB(villageAttr, i, j)
214  {
215      /* return fonction (event) { $('').innerHTML = i; }
216      -> appelé dans la fonction mère el.onclick = fonction(i);*/
217      return function()
218      {
219          hide($('carte'));
220          hide($('reset'));
221          hide($('logoJeu'));
222          show($('return'));
223          show($('ressources'));
224          show($('village'));
225
226          $('vueVillage').style.backgroundImage = $(villageAttr).style.backgroundImage;
227
228          // Affiche les informations du village
229          GenInfoVillageRPC(i,j);
230          AjoutBoutonConstrucRPC(i,j);
231      }
232  }

```

## RCP

RPC utilisé pour créer un bâtiment si l'utilisateur possède les conditions nécessaires.

```
76 // Verification qu'il a assez de ressources
77 if((( $res[0]-$qtteBouffe)>0)&& (($res[1]-$qtteBois)>0) && (($res[2]-$qttePierre)>0))
78 {
79     // Il faut modifier Le nombre de batiments et décrémenter Le nombre de ressources
    correspondant
80     $stm->closeCursor();
81     $modifRessources = "update players set
82                         food = food - :bouffe,
83                         wood = wood - :bois,
84                         stone = stone - :pierre
85                         where usr = $id";
86     if($stm = $dbh->prepare($modifRessources))
87     {
88         $etatRequete = "Ok (2)";
89         // on utilise Le param pour indiquer que L'on veut changer Les param
        précédés de ':' par La variable $xxx
90         // (elem dans la query, celui qui va La remplacer, Le type);
91         $stm->bindParam(':bouffe' , $qtteBouffe ,PDO::PARAM_INT);
92         $stm->bindParam(':bois' , $qtteBois ,PDO::PARAM_INT);
93         $stm->bindParam(':pierre' , $qttePierre ,PDO::PARAM_INT);
94         $stm->execute() or exit('<pre>'.print_r($stm->errorInfo(), true));
95         //$stm->commit();
96         $stm->closeCursor();
97
98         /*****/
99         $modifBatiment = "update map set
100                         $elemChoisi = $elemChoisi - :tpsCreat
101                         where player = $id and x = $x and y = $y";
102         if($stm = $dbh->prepare($modifBatiment))
103         {
104             $etatRequete = "Ok (3)";
105             $stm->bindParam(':tpsCreat', $tpsCreat,PDO::PARAM_INT);
106             $stm->execute();
107             //$stm->commit();
108
109             /*****/
110             $ressourceActuelle = "select food ,wood ,stone from players
111                                 where usr = ?";
112             if($stm = $dbh->prepare($ressourceActuelle))
113             {
114                 $etatRequete = "Ok (4)";
115                 // Si ça a fonctionné il renvoie Les valeurs actuelles des ressources
116                 $stm->execute(array($id));
117                 $res=$stm->fetch();
118                 for($i = 0; $i < 3; $i++)
119                 {
120                     $valueConstruc.= $res[$i]."-";
121                 }
122                 $valueConstruc.= $tpsCreat;
123             }
124             else
125             {
126                 $etatRequete = "Requete non viable (4)";
127             }
128         }
129     }
130 }
```

## HTML - CSS

```
25 <div id="page" class="container">
26   <div id="content">
27     <!-- Partie map-->
28     <div id="gamescreen">
29       <div id="ressources">
30         
31         <label id="nourriture"></label><br />
32         
33         <label id="bois"></label><br />
34         
35         <label id="pierre"></label><br />
36       </div>
37       <!-- Division pour la map-->
38       <div id="carte"><!-- La carte est générée ici. --></div>
39       <!-- Division pour la vue villages-->
40       <div id="village">
41         <div id="infoVillage"></div>
42         <div id="btnCreerBat"></div>
43         <div id="infoPersonnage"></div>
44         <div id="vueVillage"></div>
45       </div>
46
47       <!-- Partie bouton-->
48       <div id="bouton">
49         <input type="button" id="logout" value="Deconnexion"/>
50         <input type="button" id="reset" value="Reset"/>
51         <input type="button" id="return" value="Retour"/>
52       </div>
53     </div><!-- Fin de gamescreen-->
54
55     <!-- partie loginscreen-->
56     <!-- loginForm -->
57     
58     <div id="loginscreen">
59       <div id="loginForm">
60         <div id="login-triangle"></div>
61         <h2 id="login-header">Log in</h2>
62         <div id="login-container">
63           <p><input type="pseudo" id="pseudoLogin" placeholder="Pseudo"></p>
64           <p><input type="password" id="passwordLogin" placeholder="Password"></p>
65           <p>
66             <input type="button" id="login" value="Log in">
67             <input type="button" id="register" value="Register">
68           </p>
69         </div><!-- Fin de login-container-->
70       </div><!-- Fin de loginForm-->
71
72       <!-- registerForm-->
73       <div id="registerForm">
74         <div id="register-triangle"></div>
75         <h2 id="register-header">Register</h2>
76         <div id="register-container">
77           <p><input type="pseudo" id="pseudoRegister" placeholder="Pseudo"></p>
78           <p><input type="password" id="passwordRegister" placeholder="Password (1)"></p>
79           <p><input type="password" id="passwordRegisterTwo" placeholder="Password (2)"></p>
80           <p>
81             <input type="button" id="registerR" value="Register">
82             <input type="button" id="loginR" value="Retour">
83           </p>
84         </div><!-- Fin de register-container -->
85       </div><!-- Fin de register -->
86     </div><!-- Fin de loginscreen -->
87   </div><!-- Fin de content -->
```

## Résumé

- Travail effectué seul.
- Chaque semaine nous avons de nouvelles directives pour le projet, nous avons donc des deadlines.
- J'ai appris énormément de choses, la plupart du temps via des recherches personnelles.
- Utilisation de plusieurs langages de programmations donc obligation de tous les maîtriser correctement.
- Note à l'examen : 17,5/20