

Déroulement de la fabrication du jeu des énigmes
Y. Stroppa
PSB 2021

Introduction :

Ce document décrit les différentes étapes d'élaboration d'une application Web en partant de l'idée du jeu jusqu'à son développement. Il montre une démarche pour aborder ce type de problème et de construire une solution.

Voici la description du jeu.

Une simple photo d'une brochure expliquant le jeu que l'on veut construire

Quel cinéma !

À vous de retrouver le professionnel, le nombre de voitures nécessaires pour chaque cascade, et l'ordre de tournage.

INDICES

- Le seul au-dessus d'un camion n'a été réabité ni avec 1 ni 3 voitures(s), ni en 1^{er} lieu, ni par Silène.
- La cascade réalisée par Matias en 9th lieu impliquait moins de 3 voitures et n'était pas un saut ni des tonneaux.
- La poursuite avait mis en cause 5 voitures n'a été effectuée ni par Silène, ni par Johnny, ni en 1^{er} ou en 4th lieu.

Tableau de notation :

	Cascadeur				Voiture(s)				Ordre			
	Matias	Johnny	Matias	Silène	1	2	3	4	1	2	3	4
	A	B	C	D	E	F	G	H	I	J	K	L
Poursuite												
Saut de camion												
Tonneaux												
Voiture coupée en 2												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												

Coup de pouce

2 voitures ont participé à la 4th cascade

Enquêtes logiques Niveau 1/2

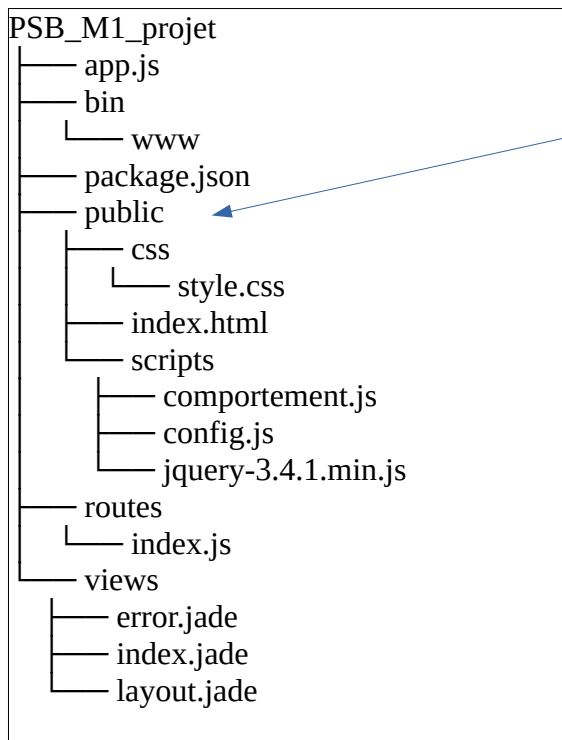
37

Etape 1 :

On va reprendre un prototype simple à partir de nodejs d'un squelette de site web à l'adresse suivante :

https://github.com/ystroppa/PSB_M1_projet

A l'aide de git, on télécharge le projet en local et on obtient l'arborescence suivante :

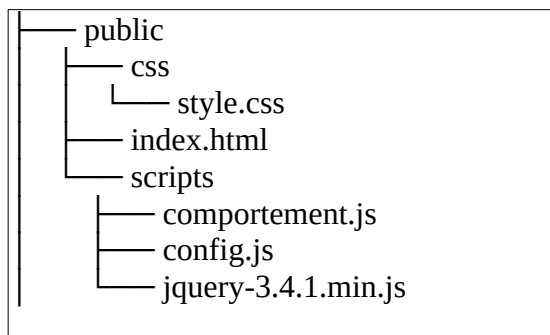


Partie public qui contient les fichiers à modifier pour notre projet. Les autres fichiers sont les paramètres de configuration de nodejs.

Pour installer le projet il faudra exécuter les commandes
npm install
npm start

Le service web démarre sur le port 4800

La partie sur laquelle



On retrouve une arborescence classique de site web que l'on pourra agrémenter en fonction de nos besoins.

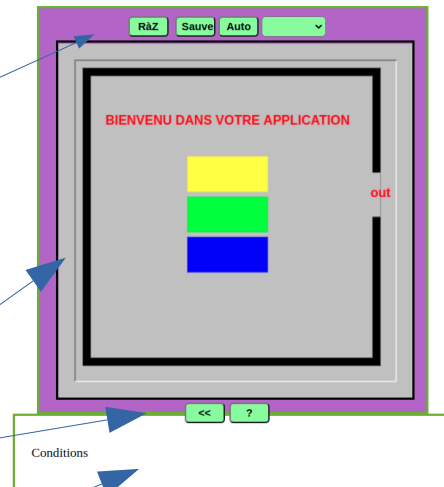
Le fichier config.js va nous servir à définir les configurations de jeux ... et le fichier comportement.js va nous servir à définir le chargement de ces configurations et la gestion du jeu.

Une fois l'application démarrée on obtient l'affichage suivant :

```
<!DOCTYPE HTML>
<html>
<head>
<title>Projet YS2020</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="/css/style.css" />
<script src="/scripts/jquery-3.4.1.min.js"></script>
</head>
<body>
<center>
<h1>Squelette du jeu 2020</h1>
</center>
<div id="conteneur" class="center">
  <div id="contenu">
    <div id="content" >
      <div id="commande_top" class="commandes">
        <input type="button" id="bt3" value="RàZ" onClick="choixRaz();"/>
        <input type="button" id="bt2" value="Sauver" onClick="choixSave();"/>
        <input type="button" id="bt1" value="Auto" onClick="depart(this)"/>
        <select name="config" id="config-select" onChange="changement_();">
        </select>
      </div>
      <div id="canv">
        <canvas id="canvas" width="440px" height="440px"></canvas>
      </div>
      <div id="commande_bottom" class="commandes">
        <input type="button" id="btReculer" value="<<" onClick="deplace('recule');"/>
        <input type="button" id="btHelp" value="?" onClick="choixRaz();"/>
      </div>
    </div>
  </div>
</div>
<div id="descriptif" class="center_texte">
  <p>
    Conditions
  </p>
</div>

<div id="piedpage">
</div>
</body>
</html>
```

Squelette du jeu 2020



Les fonctions seront à définir par la suite, seul le prototype est défini dans le fichier comportement.js

Modifier les fichiers suivants pour augmenter la taille de dessin et d'affichage
Modification des fichiers html (index.html) et css (styles.css)

```
<div id="conteneur" class="center">
  <div id="contenu">
    <div id="content" >
      <div id="commande_top" class="commandes">
        <input type="button" id="bt3" value="RàZ" onClick="choixRaz();"/>
        <input type="button" id="bt1" value="Auto" onClick="depart(this)"/>
        <select name="config" id="config-select" onChange="chargement_0;">
        </select>
      </div>
      <div id="canv">
        <canvas id="canvas" width="1000px" height="1000px"></canvas>
      </div>
      <div id="commande_bottom" class="commandes">
        <input type="button" id="btHelp" value="?" onClick="choixRaz();"/>
      </div>
    </div>
  </div>
</div>
```

On supprimer quelques éléments (boutons) et on redimensionne la zone de dessin à 1000px et 1000px.

Et pour la partie css

```
.center {
  margin: auto;
  width: 1040px;
  height: 1040px;
  border: 3px solid #73AD21;
  background-color: rgba(125, 10, 170, 0.623);
  padding: 10px 0px;
}
```

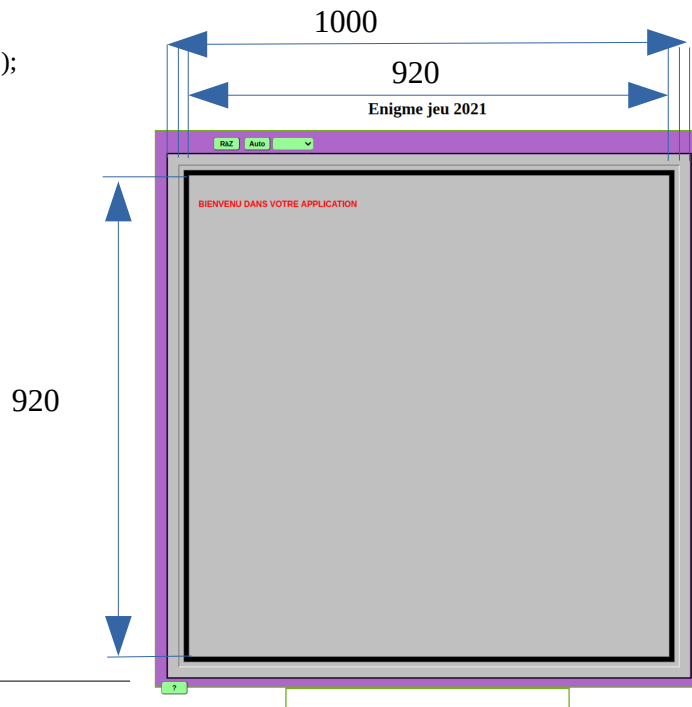
```
#canv{
  width: 1000px;
  height: 1000px;
  margin-left: auto;
  margin-right: auto;
  margin-bottom: 10px;
}
```

Vous pouvez bien sûr adapter le style.

Ensuite modification du fichier comportement.js

```
function affiche(){  
  // exemple d'affichage d'un texte dans la zone canvas  
  hh.fillStyle="red";  
  hh.fillText("BIENVENU DANS VOTRE APPLICATION",210,100);  
}
```

```
function trace(){  
  hh.font="bold 16px Arial";  
  hh.textAlign="center";  
  hh.fillStyle="silver";  
  hh.fillRect(0,0,1000,1000);  
  hh.fillStyle="black";  
  hh.fillRect(Xd+20,Yd+20,940,940);  
  hh.fillStyle="silver";  
  hh.fillRect(xmin,ymin,920,920);  
  hh.fillStyle="silver";  
  hh.fillRect(388,160,12,55);  
  hh.lineWidth=1;  
  cadre3D(20,20,980,980);  
  affiche();  
}
```



Analyse des données :

Comment définir une structure de données pour permettre de contenir la description de nos jeux.

Après analyse de l'image, que peut-on constater ?

Une partie description de l'objectif

Une partie indice pour la résolution du jeu

Une partie image

La grille du jeu

Quel cinéma !

À vous de retrouver le professionnel : nombre de voitures nécessaires pour chaque cascade, et l'ordre de tournage.

1. Le seul audace d'un camion n'a été réalisé qu'une seule fois, ni en 1^{er} lieu, ni 2^{ème}.

2. La cascade réalisée par Melas en 2^{ème} lieu impliquait moins de 3 voitures et n'était pas un seul ni des tonneaux.

3. La poursuite avait mis en cause 5 voitures n'a été effectuée ni par Silbéro, ni par Johnny, ni en 1^{er} ou en 4^{ème} lieu.

	Cascadeur	Voiture(s)	Ordre
1			
2			
3			
4			
5			

Coupe de pouce

2 voitures ont participé à la 4^{ème} cascade.

Enquêtes logiques Niveau 1/2

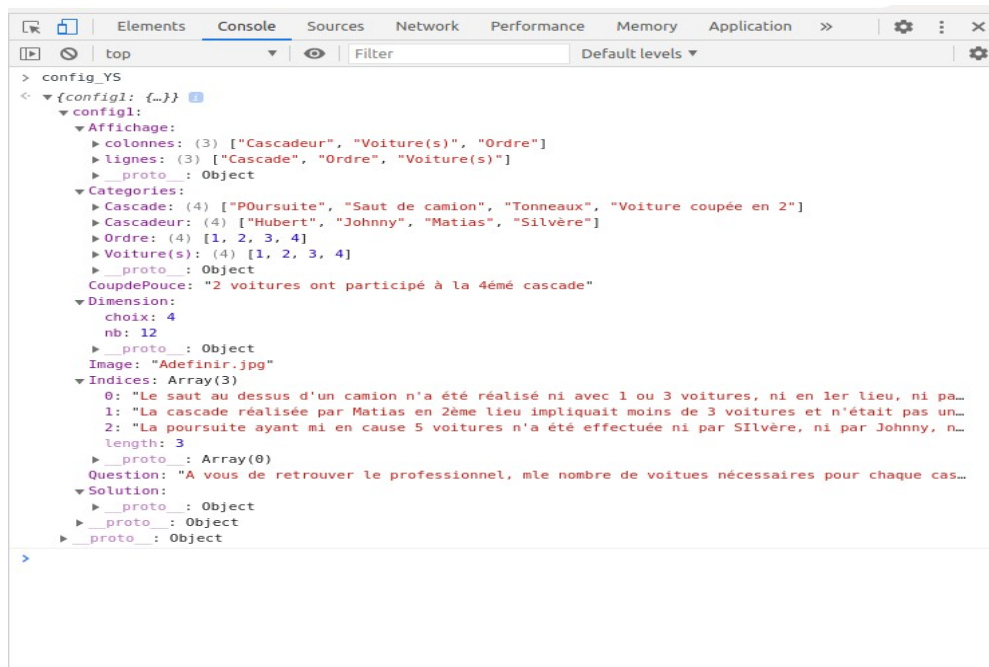
37

Création de la structure de données dans le fichier config.js

Nous allons traduire notre structure dans le fichier config.js, chaque configuration de jeu devra avoir un item : Question, Indices (un tableau car plusieurs indices possibles), Image : fichier associé à la configuration du jeu, la dimension (en nombre de colonnes et de choix dans chaque groupe), catégories permettra de définir les différents choix possibles pour la grille, et l'affichage qui permettra de définir comment disposer les éléments dans les entêtes de la grille.

```
var config_YS={
  config1: {
    Question:"A vous de retrouver le professionnel, mle nombre de voitures nécessaires pour chaque cascade et l'ordre de tournage.",
    Indices:[
      "Le saut au dessus d'un camion n'a été réalisé ni avec 1 ou 3 voitures, ni en 1er lieu, ni pas Silvère",
      "La cascade réalisée par Matias en 2ème lieu impliquait moins de 3 voitures et n'était pas un saut ni des tonneaux.",
      "La poursuite ayant mi en cause 5 voitures n'a été effectuée ni par Silvère, ni par Johnny, ni en 1er lieu ou en 4eme lieu."],
    Image:"Adefinir.jpg",
    Dimension:{nb:12,choix:4 },
    Categories:{
      Cascadeur:["Hubert","Johnny","Matias","Silvère"],
      "Voiture(s)":[1,2,3,4],
      Ordre:[1,2,3,4],
      Cascade:["POursuite","Saut de camion","Tonneaux","Voiture coupée en 2"]
    },
    Affichage:{colonnes:["Cascadeur","Voiture(s)","Ordre"],lignes:["Cascade","Ordre","Voiture(s)"]},
    CoupePouce:"2 voitures ont participé à la 4émé cascade",
    Solution:{}
  }
};
```

Une fois chargé dans le navigateur on doit pouvoir visualiser le contenu de cet objet...



Attention à déplacer les instructions dans le fichier index.html comme suit pour pouvoir indiquer la question dans la zone de description

```
</div>
<div id='descriptif' class="center_texte">
  <p>
    Conditions
  </p>
</div>
<script src='./scripts/config.js'></script>
<script src='./scripts/comportement.js'></script>
```

Définir les déclarations de variables et les fonctions de chargement suivantes :

```
// ajout pour adaatation
var entete_ligne=7;
var entete_colonne=7;
var configuration;

// appel des fonctions d'initialisation
charge_listejeux();
chargement_();

function charge_listejeux() {
  var sele = document.getElementById('config-select');
  for (var jeu in config_YS){
    var option = document.createElement("option");
    option.text = jeu;
    option.value = jeu;
    sele.add(option);
  }
}

function chargement_(){
  var sele = document.getElementById('config-select');
  var config=sele.value;
  indication=config_YS[config].Question;
  // lecture de la fonfiguration
  configuration=config_YS[config];
  choixRaz();
  $("#descriptif > p").html(indication);
  console.log("chargement de la question",indication);
}

// commandes *****
function choixRaz(){
  trace();
}
```

On aurait pu utiliser les instructions suivantes pour éviter cette dépendance
Ces instructions permettent d'indiquer les commandes à exécuter lorsque le document est chargé ... à ajouter dans le fichier comportement.js

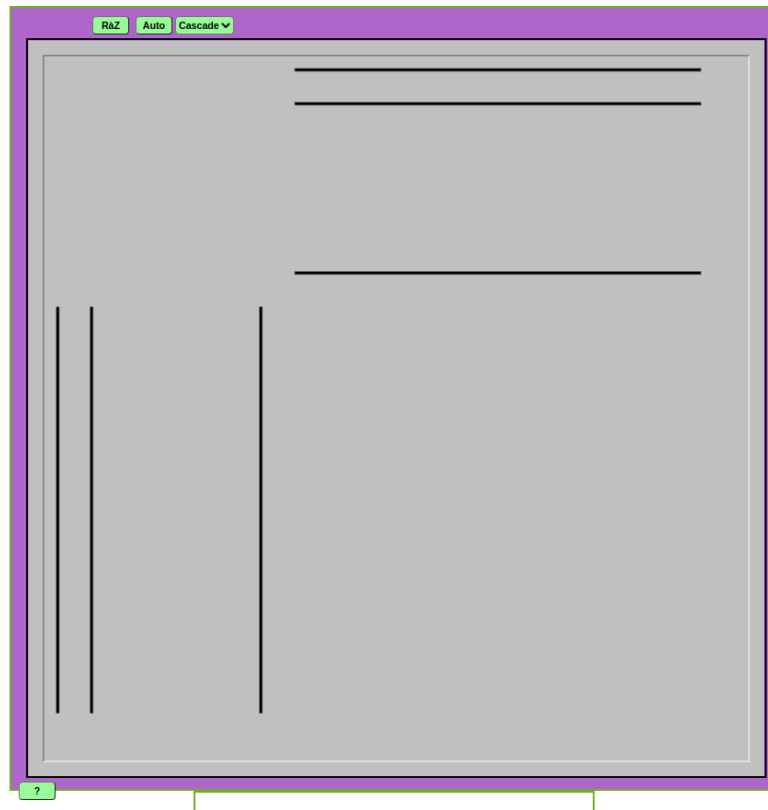
```
document.addEventListener("DOMContentLoaded", function(event) {
  // appel des fonctions d'initialisation
  charge_listejeux();
  chargement_();
});
```


On va s'occuper de la fonction trace pour les différentes constructions du maillage de notre jeu

D'abord on s'occupe des entêtes de lignes et de colonnes

```
function trace_entetes_lignes() {  
  // trace des entetes de colonnes  
  hh.strokeStyle="black";  
  hh.lineWidth = 4;  
  let y=[0+ymin,1*tl+ymin,(entete_-1)*tl+ymin];  
  for (let yi of y) {  
    drawLine(entete_*tc+xmin,yi,nb_cases*tc+xmin,yi);  
  }  
  let x=[0+xmin,1*tc+xmin,(entete_-1)*tc+xmin];  
  for (let xi of x) {  
    drawLine(xi,entete_*tl+ymin,xi,nb_cases*tl+ymin);  
  }  
}
```

Enigme jeu 2021



On construit ensuite le tracé des autres lignes des catégories

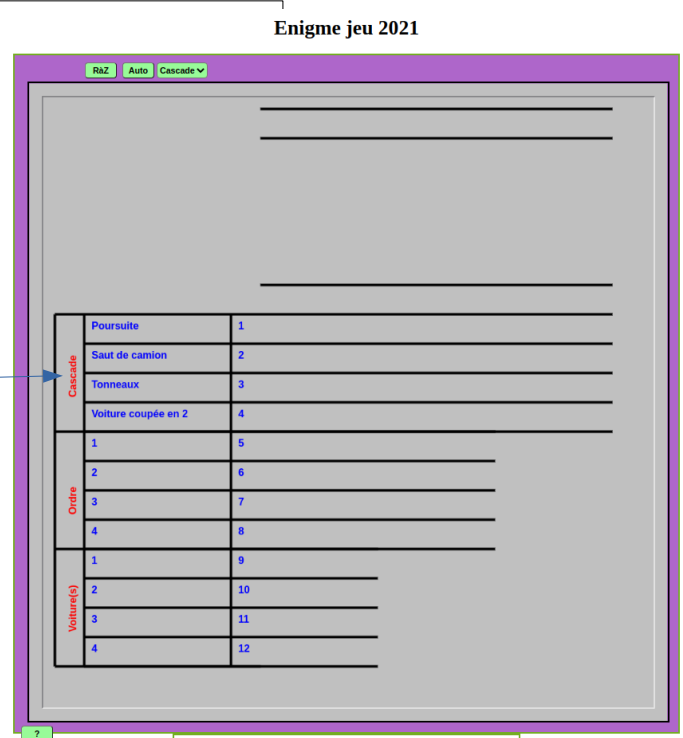
```
function trace_lignes() {
  // trace des entetes de colonnes
  hh.strokeStyle="black";
  hh.lineWidth = 4;
  // tracé des lignes par rubriques
  let fin_ligne=nb_cases*tc+ymin, debut_ligne=0+ymin, ligne_courante=entete_*tl+ymin;
  let numero=1;
  for (let cat of configuration.Affichage.Lignes)
  {
    drawLine(debut_ligne,ligne_courante,fin_ligne,ligne_courante);
    for (let elem of configuration.Categories[cat]) {
      // ecrire l'element
      hh.fillStyle="blue";
      hh.textAlign = "left";
      hh.fillText(elem,debut_ligne+1.25*tc, ligne_courante+0.5*tl);
      hh.fillText(numero,debut_ligne+6.25*tc, ligne_courante+0.5*tl);
      // trace ligne
      ligne_courante+=tl;
      numero+=1;
      drawLine(debut_ligne+1*tl,ligne_courante,fin_ligne,ligne_courante);
    }
    fin_ligne-=configuration.Dimension.choix*tl;
  }
  drawLine(debut_ligne,ligne_courante,fin_ligne,ligne_courante);
}
```

Enigme jeu 2021

R&Z
Auto
Cascade ▾

	Poursuite	1	
	Saut de camion	2	
	Tonneaux	3	
	Voiture coupée en 2	4	
	1	5	
	2	6	
	3	7	
	4	8	
	1	9	
	2	10	
	3	11	
	4	12	

```
function trace_lignes() {
  // trace des entetes de colonnes
  hh.strokeStyle="black";
  hh.lineWidth = 4;
  // tracé des lignes par rubriques
  let fin_ligne=nb_cases*tc+ymin, debut_ligne=0+ymin, ligne_courante=ent
  let numero=1;
  for (let categ of configuration.Affichage.Lignes)
  {
    drawLine(debut_ligne,ligne_courante,fin_ligne,ligne_courante);
    hh.fillStyle="red";
    hh.save();
    hh.translate(debut_ligne+0.5*tc, ligne_courante+5*tl);
    hh.rotate(-Math.PI/2);
    hh.textAlign = "left";
    hh.fillText(categ, 100, 10);
    hh.restore();
    for (let elem of configuration.Categories[categ]) {
      // ecrire l'element
      hh.fillStyle="blue";
      hh.textAlign = "left";
      hh.fillText(elem,debut_ligne+1.25*tc, ligne_courante+0.5*tl);
      hh.fillText(numero,debut_ligne+6.25*tc, ligne_courante+0.5*tl);
      // trace ligne
      ligne_courante+=tl;
      numero+=1;
      drawLine(debut_ligne+1*tl,ligne_courante,fin_ligne,ligne_courante);
    }
    fin_ligne-=configuration.Dimension.choix*tl;
  }
  drawLine(debut_ligne,ligne_courante,fin_ligne,ligne_courante);
}
```



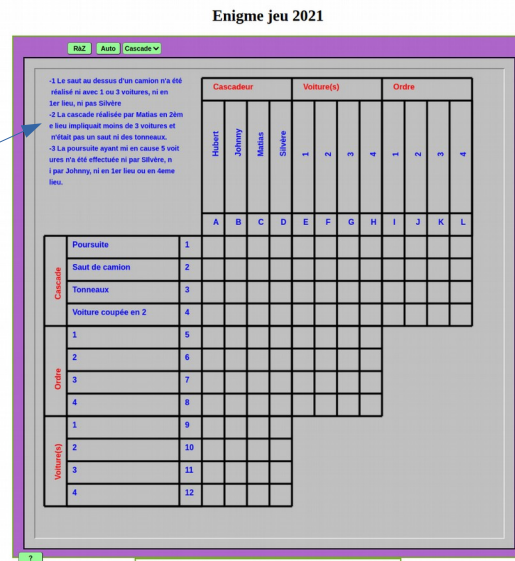
ce qui nous donne la partie à rajouter à la suite du code de la fonction

Enigme jeu 2021

[illegible]

IL faut ajouter le texte avec les indices pour chaque configuration :

```
function ecriture_indices() {
  hh.fillStyle="blue";
  hh.font="bold 14px Arial";
  var ligne=0;
  let nb_caract=40;
  for (let ind of configuration.Indices) {
    // il faut découper chaque ligne de 30 caractères
    let nb_parties=ind.length/nb_caract;
    let debut=0;
    for (let i=0; i<nb_parties; i++) {
      hh.fillText(ind.substr(debut,nb_caract), 50, 50+ligne);
      ligne+=tl*.5;
      debut+=nb_caract;
    }
  }
}
```



On va pouvoir tester avec une deuxième configuration

```
Attractions: {
  Question:"Frissons garantis pour Ludivine et Frédéric qui on passé une journée au parx EffroiLand ...",
  Indices:[
    "-1 La maison du diable n'a pas été visitée apès 50min de file, ni en 2min30, ni en 3e lieu",
    "-2 L'attente au Miroir aux alouettes a été deux fois plus longue qu'à la première attraction, visitée en 2min",
    "-3 Les mines de Salomon, visitées en 1min30 après moins de 50min d'attente, ,étaient ni la 3e ni la 5e
attraction.",
    "-4 Pour la 4e attraction, visitée en une minute de plus que le Tunnelde l'épouvante, Ludivine et Frédéric on
tattendu 40min."],
  Image:"Adefinir.jpg",
  Dimension:{nb:15,choix:5 },
  Categories:{
    Duree:["1 min","1 min 30","2 min","2 min 30","3 min"],
    Ordre:[1,2,3,4,5],
    Attente: ["20 min","30 min","40 min","50 min","60 min"],
    Attraction:["La maison du diable","Les mines de salomon","Le miroir aux alouettes","Le temple de
Sésostris","Le tunnel de l'épouvante"]
  },
  Affichage:{Colonnes:["Duree","Ordre","Attente"],Lignes:["Attraction","Attente","Ordre"]},
  CoupdePouce:"",
  Solution:{}
}
```

Ce qui nous donne

Enigme jeu 2021

RàZAutoAttraction

-1 La maison du diable n'a pas été visitée après 50min de file, ni en 2min30, ni en 3e lieu

-2 L'attente au Miroir aux alouettes a été deux fois plus longue qu'à la première attraction, visitée en 2min

-3 Les mines de Salomon, visitées en 1min30 après moins de 50min d'attente, 'étaient ni la 3e ni la 5e attraction.

-4 Pour la 4e attraction, visitée en une minute de plus que le Tunnel de l'épouvante, Ludivine et Frédéric ont attendu 40 min.

		Duree					Ordre					Attente				
		1 min	1 min 30	2 min	2 min 30	3 min	1	2	3	4	5	20 min	30 min	40 min	50 min	60 min
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Attraction	La maison du diable	1														
	Les mines de salomon	2														
	Le miroir aux alouettes	3														
	Le temple de Sésostris	4														
	Le tunnel de l'épouvante	5														
Attente	20 min	6														
	30 min	7														
	40 min	8														
	50 min	9														
	60 min	10														
Ordre	1	11														
	2	12														
	3	13														
	4	14														
	5	15														

?

La gestion du jeu et de l'interaction avec l'utilisateur

On va ajouter à la variable fenetre positionnée sur l'objet Canvas un module d'écoute .

```
fenetre.addEventListener('click', function(evt) {
    var x = evt.pageX;
    var y = evt.pageY;
    var node = evt.target;
    while (node) {
        x -= node.offsetLeft - node.scrollLeft;
        y -= node.offsetTop - node.scrollTop;
        node = node.offsetParent;
    }
    Xs=x-3;          Ys=y-3;
    I=1+Math.floor((Xs-xmin)/tc);
    J=Math.floor((Ys-ymin)/tl);
    console.log(X,Y);
});
```

Comment faire pour permettre la modifications des cases en fonction du choix de l'utilisateur ?

On va construire un objet qui va représenter le maillage de notre jeu et chaque case sera un élément de ce maillage et aura 3 états lorsque la case est dans la partie modifiable du jeu

- 0 : pas de marquer
- 1 : un X pour indiquer que la cellule n'est pas retenue
- 2 : pour indiquer que l'on retient la cellule

On ajoute l'initialisation de cette ,nouvelle variable pour ne remplir que les cases modifiables

```
function choixRaz() {
    maillage=[];

    zone_dessin = document.getElementById('canvas');
    nb_cases=configuration.Dimension.nb+entete_;
    let choix = configuration.Dimension.choix;
    for (let i=0;i<nb_cases;i++){
        maillage[i]=[];
    }
    tl=(zone_dessin.height-2*xmin)/(nb_cases+1);
    tc=(zone_dessin.width-2*ymin)/(nb_cases+1);
    let fin_ligne=nb_cases;
    for (let i=0;i<nb_cases;i++){
        if (i>7)
            if ((i-7)%choix==0 && i!=0){
                fin_ligne-=choix;
            }
        for (let j=0;j<nb_cases;j++){
            maillage[i][j]=0;
            if (i<7) maillage[i][j]=-1;
            if (j<7) maillage[i][j]=-1;
            if (j>=fin_ligne) maillage[i][j]=-1;
        }
    }
    trace();
}
```

Pour prendre en compte la structure particulière du jeu.

On va ajouter une nouvelle fonction pour le dessin des cases et la relier à l'événement click sur le jeu

....

Ce qui nous donne

```
function redessine_maillage() {
  hh.font="bold 20px Arial";
  for (let i=0;i<nb_cases;i++){
    for (let j=0;j<nb_cases;j++){
      let posit_x=(i)*tc+xmin;
      let posit_y=(j+1)*tl+ymin;
      switch (maillage[i][j]) {
        case 0:
          hh.fillStyle="silver";
          hh.fillRect(posit_x+5,posit_y+5-tl,tc-10,tl-10);
          hh.fillStyle="blue";
          hh.fillText("",posit_x+tc/4,posit_y-tl/4);
          break;
        case 1:
          hh.fillStyle="silver";
          hh.fillRect(posit_x+5,posit_y-tl+5,tc-10,tl-10);
          hh.fillStyle="red";
          hh.fillText("X",posit_x+tc/4,posit_y-tl/4);
          break;
        case 2:
          hh.fillStyle="silver";
          hh.fillRect(posit_x+5,posit_y+5-tl,tc-10,tl-10);
          hh.fillStyle="blue";
          hh.fillText("O",posit_x+tc/4,posit_y-tl/4);
          break;
      }
    }
  }
}
```

Et le lien avec la gestion d'événements

```
// 1 evenementiel
fenetre.addEventListener('click', function(evt) {
  var x = evt.pageX;
  var y = evt.pageY;
  var node = evt.target;
  while (node) {
    x -= node.offsetLeft - node.scrollLeft;
    y -= node.offsetTop - node.scrollTop;
    node = node.offsetParent;
  }
  Xs=x-3;          Ys=y-3;
  I=Math.floor((Xs-xmin)/tc);
  J=Math.floor((Ys-ymin)/tl);
  if (maillage[I][J]!=-1) {
    // il faut changer l'état du maillage et redessiner
    console.log(maillage[I][J]);
    switch (maillage[I][J]) {
      case 0:
        maillage[I][J]=1;
        break;
      case 1:
        maillage[I][J]=2;
        break;
      case 2:
        maillage[I][J]=0;
        break;
    }
    redessine_maillage();
  }
});
```


Pour aller plus loin :

Veuillez charger les configurations de jeux suivantes

On pourra ajouter également la gestion des solutions sous quelle forme peut-on définir les solutions dans la partie configuration et comment éviter que la solution soit ramener du côté du client dans la fichier config.js, peut-con imaginer que la solution au problème se trouve du côté du serveur ou du côté d'un cache mais que l'accès soit conditionné

Egalement dans la saisie des résultats, comment ouvrir une boîte de dialogue qui permet à l'utilisateur de saisir sa réponse et de la valider auprès d'un service de vérification