

Labyrinthe of the Dead :

Choix du projet :

Dans un premier temps, nous voulions faire un projet qui nous servirait dans notre propre intérêt, on s'était donc mis d'accord sur un logiciel de mathématiques qui regrouperait plein de petit programme, ce qui nous aurait permis de réviser pour le bac. Mais lorsqu'on a commencé à réaliser des petits jeux en classe comme « la bataille naval », « plus ou moins », ou « Mastermind », ça nous a donné envie d'en réaliser un pour notre projet final.

L'idée de faire un labyrinthe nous est venue lorsqu'on s'est retrouvé à réaliser des Labyrinthes sur des feuilles lors d'un cour d'ISN.

Finalement notre but final a évolué, et nous ne voulions plus faire un logiciel pour nous mais un jeu qu'on pourrait faire partager avec la classe et nos entourages.

Répartition des taches :

Au commencement du projet nous nous étions mis d'accord sur le fait que nous fassions le projet ensemble sans partager les tâches. Nous avons donc commencé à programmer le labyrinthe à trois, mais nous nous sommes très vite rendu compte que ça ne pouvait pas marcher car il nous manquerait de temps et les tensions au sein du groupe se faisaient sentir. Nous avons finalement décidé de se répartir les tâches et de se faire confiance les uns aux autres, cela nous permettait également de travailler individuellement chez soi.

La répartition du travail s'est faite comme ça :

- Ludovic = Tous ce qui était en rapport avec le java script du Labyrinthe.
- David = La page de règle du jeu.
- Florian = CSS de la page de règle du jeu et de la page du Labyrinthe.

Mais cela était juste un plan pour s'orienter, car au final tout le monde a participé un peu à toutes les taches, et nous nous sommes beaucoup aidés les uns, les autres.

Réalisation du projet :

La réalisation du tableau s'est fait rapidement, il n'y a eu aucun problème à ce niveau-là.

Une fois le tableau fini, les choses se sont compliquées.

Il nous a fallu créer un joueur « un carré », qui est capable de se déplacer dans le tableau, pour cela,

<pre><INPUT type="hidden" name="positionX"></pre>	nous avons créé 4 « hidden » qui vont nous permettre de définir une zone de formulaire invisible à l'écran, nous pourrons donc y stocker les informations sur la position des joueurs.
<pre><INPUT type="hidden" name="positionY"></pre>	
<pre><INPUT type="hidden" name="positionT"></pre>	
<pre><INPUT type="hidden" name="positionS"></pre>	

Ensuite nous avons réalisé la première fonction :

```
function départ() //position des joueurs au départ
{
    créemur()

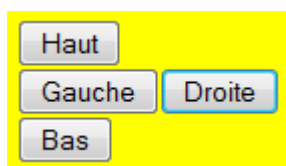
    document.getElementById("0,3").style.backgroundColor = CJ1 //position départ joueur1
    document.getElementById("f0,3").color= CJ1 //position départ joueur1
    document.form.positionX.value = 3 //position départ joueur1
    document.form.positionY.value = 0 //position départ joueur1

    document.getElementById("0,4").style.backgroundColor = CJ2 //position départ joueur2
    document.getElementById("f0,4").color= CJ2 //position départ joueur2
    document.form.positionT.value = 0 //position départ joueur2
    document.form.positionS.value = 0 //position départ joueur2

    document.getElementById(positionFin).style.backgroundColor = "#FF00FF" //Position fin
    document.getElementById("f"+positionFin).color = "#FF00FF" //position fin
}
```

Cette fonction permet d'afficher la position de départ des joueurs en changeant la couleur de la case où se trouveront les joueurs et en affectant les coordonnées de départ aux différents « hidden » vu au-dessus.

Dans la même occasion, cette fonction permet de définir la couleur de la case d'arriver.



Une fois les joueurs créés et placés au point de départ, il a fallu trouver comment les faire bouger.

Pour cela nous avons d'abord pensé à des boutons sur lesquels il faudrait cliquer pour permettre le déplacement du joueur (voir photo si gauche). Mais cette méthode a posé problème lorsqu'on a voulu faire un mode multi-joueurs

car on ne pouvait pas cliquer sur deux boutons à la fois tandis qu'on avait qu'une souris, donc j'ai mis en place un système pour que lorsqu'on appuie sur une touche du clavier, le joueur se déplace. J'ai repris les touches habituelles des différents jeux qu'on peut trouver sur internet :

Premier joueur : les 4 flèches directionnelles.

Deuxième joueur : les 4 touches « z,q,s,d ».

Ensuite nous avons créé une fonction « move » pour le joueur 1 et « nove » pour le joueur 2. Ces deux fonctions ont le même rôle qui est de récupérer la touche qui a été appuyée, vérifier si la prochaine case est libre (grâce à la fonction check), et si c'est le cas de lancer la nouvelle fonction (afficher) qui va permettre le déplacement.

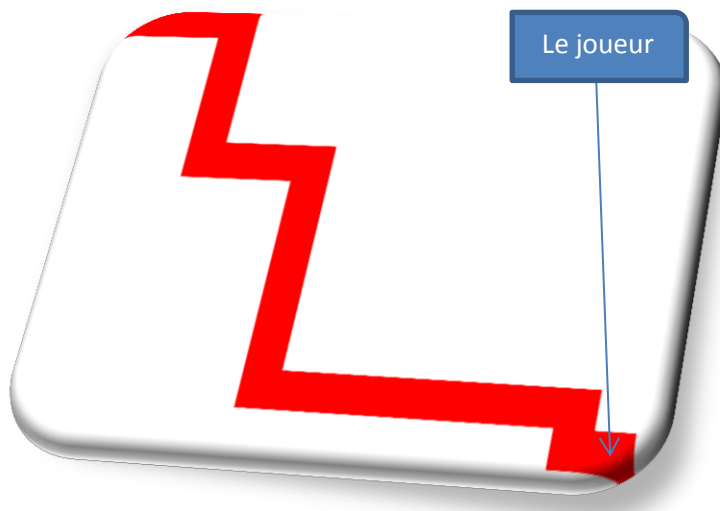
```
function move(e) // joueur 1
{
    y = document.form.positionY.value*1
    x = document.form.positionX.value*1
    if (e == 1 && check(y+1,x))
    {
        afficher(y+1,x)
    }
    if (e == 2 && check(y,x+1))
    {
        afficher(y,x+1)
    }
    if (e == 3 && check(y-1,x))
    {
        afficher(y-1,x)
    }
    if (e == 4 && check(y,x-1))
    {
        afficher(y,x-1)
    }
}
```

La fonction (afficher) va donc permettre de récupérer les nouvelles coordonnées du joueur et va faire à peu près la même chose que la fonction « départ » vu au début mais avec les nouvelles coordonnées.

Cette fonction active également une autre fonction « effacer ».

```
function afficher(Y,X) //joueur1    Affiche le carré du joueur.
{
  effacer(document.form.positionY.value,document.form.positionX.value)
  document.getElementById(Y+", "+X).style.backgroundColor = CJ1
  document.getElementById("f"+Y+", "+X).color=CJ1
  document.form.positionX.value = X
  document.form.positionY.value = Y
  vérifier()
}
```

Nous avons réussi finalement à faire déplacer le jouer. Mais un nouveau problème est apparu :



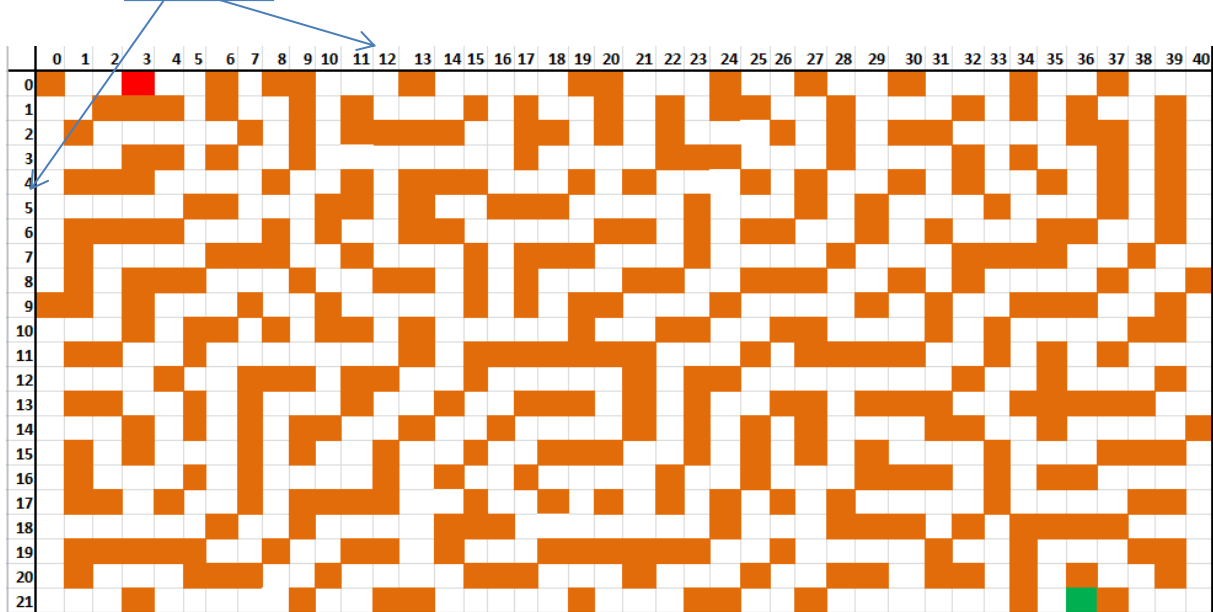
Nous avons constaté que le joueur laissait derrière lui la trace de son passage, donc ça ne correspondait pas à ce qu'on attendait. Nous voulions seulement un carré qui bouge sur le tableau et non « une grosse trace ». Il a donc fallu régler ce problème.

Pour cela, nous avons créé une nouvelle fonction « Effacer » qui a pour but qu'à chaque fois qu'on appuie sur une touche de récupérer les coordonnées où se trouve le joueur et de mettre la case correspondante au coordonné de la couleur blanche pour supprimer la couleur du passage du joueur.

```
function effacer(Y,X) //joueur1
{
  document.getElementById(Y+", "+X).style.backgroundColor = Trace1
  document.getElementById("f"+Y+", "+X).color = Trace1
}
```

Les mouvements des joueurs ont pu être enfin opérationnels ! Mais notre labyrinthe n'a pas de mur, il a donc fallu remédier à ça.

On a alors réalisé une maquette de notre labyrinthe sur un fichier excel qui allait nous permettre d'avoir les coordonnées de tous les murs à placer.



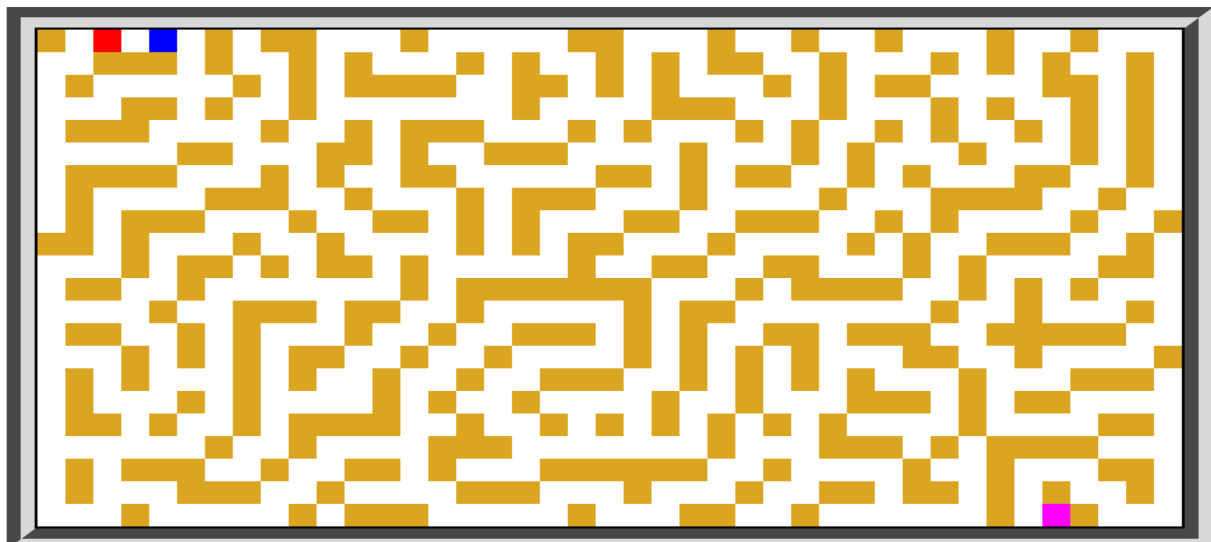
Ensuite comme on avait les coordonnées, ils nous a simplement fallu les noter pour enfin voir apparaître ces murs.

C'est à ce moment-là ou nous avons eu l'idée de mettre un mode multi-joueurs, ce qui veut dire un deuxième personnage qui serait capable de bouger en même temps que le premier. Nous nous sommes dit que cela pourrait rendre notre jeu plus intéressant et permettrait de nous rapprocher de notre objectif qui était de créer un jeu multi-joueurs.

J'ai donc pris le temps de tous reprendre pour adapter ce deuxième personnage.

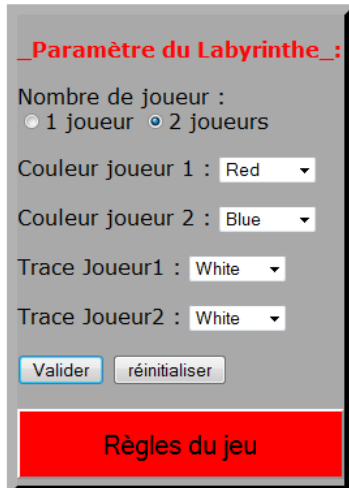
Pendant ce temps, David a commencé la page de présentation des règles du jeu, quand à Florian qui a fait une partie des CSS de cette page. Et par la suite j'ai entamé les CSS de la page de jeu.

Ceci a été fait relativement vite et le jeu commençait enfin à prendre forme.



Néanmoins on a décidé d'apporter d'autres éléments dans notre projet, on a donc eu l'idée de mettre en place un système de personnalisation du Labyrinthe où nous pourrions choisir si on voulait jouer tous seul ou en multi-joueurs, la couleur des joueurs et si on voulait faire apparaître des traces derrière le joueur.

J'ai décidé de m'occuper de cette partie pendant que Florian et David s'occupaient de relier les différentes pages entre elles.



J'ai donc commencé par créer tous les boutons et les organiser à ma façon grâce au CSS.

Ensuite je me suis lancé à programmer le rôle de chaque bouton en JavaScript.

Pour avoir le choix entre 1 ou 2 joueurs, j'ai utilisé 2 conditions « If » :

Premier If :

-Il récupère les coordonnées de la cellule « 0,2 » et la colore en blanc (couleur de fond du labyrinthe).

-Il récupère les coordonnées de la cellule « 0,3 » et la colore de la couleur du joueur 1.

Et il affecte les coordonnées du point de départ du joueur 1 aux hidden correspondant (position X et Y).

-Il récupère les coordonnées de la cellule « 0,4 » et la colore en blanc pour pas qu'on la voit.

Et il affecte des mauvaises coordonnées au joueur2 (coordonnées d'un mur).

Deuxième If :

```
if(document.getElementById('OneJoueur').checked) {

document.getElementById("0,2").style.backgroundColor = "White"
document.getElementById("f0,2").color="White"

document.getElementById("0,3").style.backgroundColor = CJ1
document.getElementById("f0,3").color=CJ1
document.form.positionX.value = 3
document.form.positionY.value = 0

document.getElementById("0,4").style.backgroundColor = "White"
document.getElementById("f0,4").color="White"
document.form.positionT.value = 0
document.form.positionS.value = 0

}

if(document.getElementById('TwoJoueurs').checked) {

document.getElementById("0,2").style.backgroundColor = CJ1
document.getElementById("f0,2").color=CJ1
document.form.positionX.value = 2
document.form.positionY.value = 0

document.getElementById("0,4").style.backgroundColor = CJ2
document.getElementById("f0,4").color=CJ2
document.form.positionT.value = 4
document.form.positionS.value = 0

document.getElementById("0,3").style.backgroundColor = "White"
document.getElementById("f0,3").color="White"

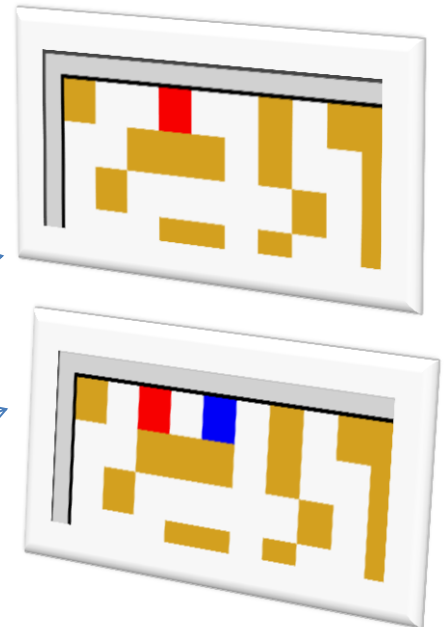
}
}
```

Il procède de la même façon, mais en affectant au joueur 2 de bonnes coordonnées.

Selon le mode de jeu, j'ai placé différemment les positions de départ des joueurs pour que ça soit plus esthétique et organisé.

Position lorsqu'il
y a 1 joueur

Position lorsqu'il
y a 2 joueurs



Ensuite je me suis occupé de tout ce qui était choix des couleurs pour les joueurs et les tracés.

```
var CJ1 = "red";  
var CJ2 = "White" ;  
var Trace1 = "white";  
var Trace2 = "White";
```

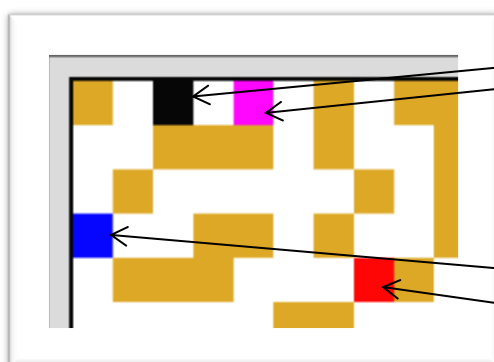
J'ai fait ceci à l'aide de 4 variables dans lesquels sont stockée la couleur choisie « en chaine de caractère ».

Ensuite pour affecter la couleur choisie à la variable, j'ai créé une fonction qui se lance lorsqu'on appuie sur le bouton « Valider ».

Le fonctionnement est simple, il récupère juste la couleur en chaine de caractère choisi dans les boutons « select » placés sur la page html et l'injecte à la variable correspondante.

```
function action1() { //Configurer le bouton  
  
var Val1 = document.getElementById("1J").value;  
CJ1 = Val1;  
  
var Val2 = document.getElementById("2J").value;  
CJ2 = Val2;  
  
var Val3 = document.getElementById("Trace1").value;  
Trace1 = Val3;  
  
var Val4 = document.getElementById("Trace2").value;  
Trace2 = Val4;
```

Néanmoins cette méthode présente un problème que je n'ai pas su résoudre. Lorsqu'on se déplace dans le labyrinthe et qu'on appuie sur le bouton « valider », les cases précédentes colorées ne disparaissent pas.



Nouveaux
emplacements des
joueurs

Anciens emplacements
des joueurs

Donc pour remédier à ce problème, j'ai ajouté un bouton « reset » qui permet de réinitialiser la page et donc de faire disparaître toutes ces traces d'anciens emplacements.

Voilà enfin le projet fini !

Idée pour l'avenir du Labyrinthe :

Nous avons encore quelques idées pour améliorer notre Labyrinthe mais nous n'avons pas eu le temps de les réaliser. Mais je compte finir ce Labyrinthe et réaliser ses idées même si l'oral est déjà passé car je n'ai pas envie de laisser ce projet à ce stade et ça m'entraînera pour mes études à venir dans l'informatique.

Quelques idées :

- Création d'un système de point.
- Création d'un système de temps (les joueurs auraient un temps pour finir le labyrinthe sinon retour à 0 des joueurs et du chrono).
- Création d'un système de record de temps (Je compte le réaliser en php).
- Création d'un système permettant au joueur de personnaliser le plateau de jeu à leurs goûts (couleur des murs, du fond, des contours).

Nous avons commencé à le réaliser, mais un manque de technique de notre part et de temps nous a bloqués.

- Création d'un algorithme qui permettrait de réaliser des labyrinthes aléatoirement (Nous avons également voulu le faire, mais encore un manque de technique de notre part).

Conclusion :

Nous avons créé un jeu simple qui correspondait à nos attentes. Nous avons essayé de mettre en pratique le maximum de connaissances que nous avons acquies le long de cette année (Html, CSS et JavaScript ont été nos 3 langages utilisés).

J'espère que vous apprécierez ce projet autant que nous.