Informatique : Travail de Fin d'année – 4TTr Année académique : 2020- 2021

pac-man

Leseure enzo



DÉFINITION DU PROJET

Le jeu Pac-man consiste à faire le plus grand score en ramassant le plus de pièces et en esquivent les squelettes qui essayeront de vous tuer en vous fonçant dessus.

Comment jouer?

Comme explique plus haut le but est de faire le plus haut score.il y a différente manière de gagner des point, la principale est de manger les pièces éparpillé partout dans le labyrinthe. Quand vous avez mangé toute les pièces du labyrinthe, Pac-man réapparaît a son point de départ et les squelettes retourne dans leur lieux de réapparition avant que toutes les pièces réapparaissent. Vous pouvez aussi gagner des point en tuant les squelettes. Pour tuer les squelettes il faut d'abord ramasser l'épée qui vous permettra de les tuer pendant un délais de 20 seconde pendant lequel les squelettes seront désarmé. Après avoir été tuer les squelettes réapparaissent dans leur lieux de réapparition et se remettent a vous pourchasser. La dernière façon de gagner des point est de ramasser des item donnant plus ดน moins de point selon le niveaux dans leguel vous vous trouver.

Le héros avance en continu, pour changer la direction vers laquelle il va aller il faut utiliser les touches directionnelle du clavier. les squelettes essayeront de vous tuer en vous poursuivant et si il vous touche vous mourrez, Vous pouvez mourir trois fois. vous pouvez regagner une vie en ramassant une potion.

DESCRIPTION DES INTERFACES ET CAS D'UTILISATION



Menu principal :

Le menu principal reprend 3 boutons

PLAY:permet d'accéder à la page du jeu

CRÉDITS : permet d'accéder à la page crédits

• **RÈGLES** : permet d'accéder à la page règles



Menu Règles

La page contient les règles du jeu.

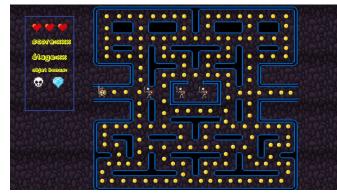
Un bouton permet de revenir à l'écran d'accueil.



Menu Crédits

le menu crédits reprend la personne qui a fait le projet et l'année ou il a été fait

Un bouton permet de revenir à l'écran d'accueil.



interface du jeu

- 1.l'interface contient une zone qui se situe en heut à gauche où le nombre de vie et le score sont affiché.
- 2.l'interface contient le labyrinthe dans lequel Pacman va essayer de ramasser le plus de point.si vous passez dans le tunnel de gauche Pac-man ressortira par le tunnel de droite.

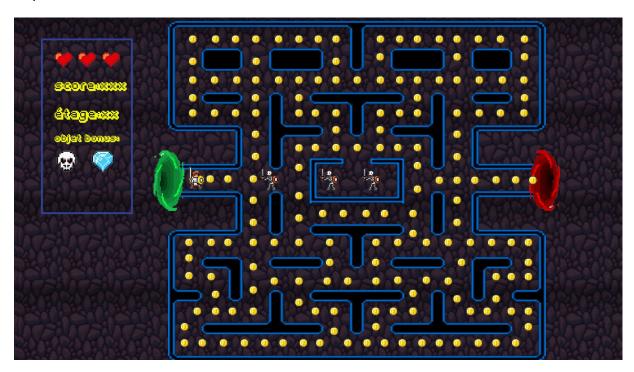


Fin de partie

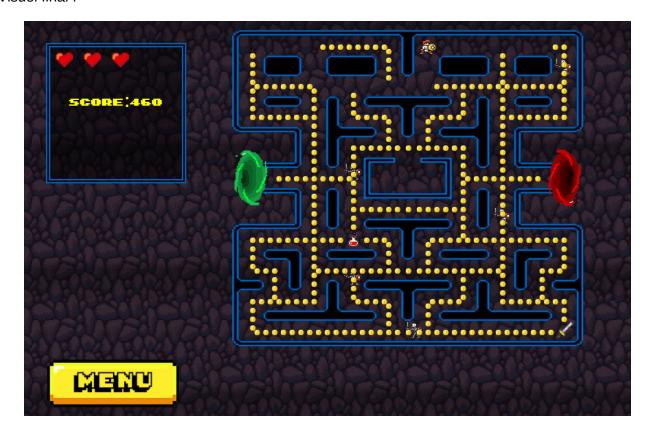
votre score apparaît à l'écran ainsi qu'un bouton pour rejouer une partie et un bouton pour retourner au menu principal

Visuel prévu et visuel final :

Visuel prévu :



Visuel final:



Tache a réaliser	Temps estimé	Temps passé
Création structure du site	3h	3h
Recherche d'image et construction des éléments graphiques.	3h	7h
Création de la map + lieu d'apparition des ennemi	3h	6h
Mouvement du personnage	3h	4h
Mouvement des ennemis	3h	4h
Effets des objets	6h	2h
Compteur de score en temps réel	3h	1h
Système de vie	2h	2h
Création des ennemis	2h	4h
Système de récolte de points	5h	3h
Système de mort du personnage	2h	4h
Typographie	1h	1h
Écran perdu	1h	2h
Création héros	2h	3h
test	6h	8h
débugs	2h	4h
animation	3h	3h

Code:

Index.html:

```
<!DOCTYPE html>
<html>
         <head>
                   k rel="stylesheet" type="text/css" href="style.css">
                   <title>menu</title>
          </head>
         <body class="body1">
                   <div id="buttonbox2">
                             <input onclick="parametre()" id="optionbutton" type="image" src="img/engrenage.png">
                   </div>
                   <div id="gamename">
                             <img id="namegame2" src="img/name.png">
                   </div><div id="buttonbox">
                             <a href="regles.html" ><input class="buttonmenu" type="image" src="img/button(play).png"></a>
                             <a href="credits.html"><input class="buttonmenu" type="image" src="img/buton(credits).png"></a>
                   </div>
                   <div id="fond">
                                      <div id="boxbutton3">
                                                <input onclick="parametre2()" id="cross" type="image" src="img/cross.png">
                                      </div>
                   </div>
                   <script type="text/javascript" src="script2.js"></script>
         </body>
</html>
```

credits.html: <!DOCTYPE html> <html> <head> k rel="stylesheet" type="text/css" href="style.css"> <title>crédits</title> </head> <body class="body1"> <div id="buttonbox"> <input class="buttonmenu" type="image" src="img/button(MENU).png"> </div> <div class="boxtxt"> <h2 id="ttrregles">CREDITS DU JEU</h2>
 Créateur : Leseure Enzo Classe : 4TTr Année : 2020-2021

</div>

</body>

</html>

<script type="text/javascript" src="script2.js"></script>

```
regles.html:
<!DOCTYPE html>
<html>
         <head>
                   k rel="stylesheet" type="text/css" href="style.css">
                   <title>
                            règles</title>
         </head>
         <body class="body1">
                   <div class="boxtxt">
                                      <h2 id="ttrregles">REGLES DU JEU</h2>
                                     Le jeu Pac-man consiste a faire le plus grand score en ramassant le plus de
pieces d'ors et en esquivent les squelette qui essayeront de vous tuer en vous fonçant dessus avec leur epees.
                                      </div>
                   <div id="buttonbox4">
                            <a href="index.html"><input class="buttonmenu" type="image" src="img/button(MENU).png"></a>
                   </div>
                   <div id="buttonbox5">
                            <\!a\,href="game.html"><\!input\,class="buttonmenu"\,type="image"\,src="img/button(play).png"><\!/a>
                   </div>
                   <script type="text/javascript" src="script2.js"></script>
         </body>
</html>
```

```
game.html:
<!DOCTYPE html>
<html>
<head>
         k rel="stylesheet" type="text/css" href="style.css">
         <title>jeu</title>
</head>
<body id="body2">
         <audio id="myAudio">
                   <source type="audio/ogg">
                   <source type="audio/mpeg">
         </audio>
         <input onclick="bbb()" id="optionbutton2" type="image" src="img/son_pasactif.png">
         <div id="fin">
         <div id="cadre">
                            <div id="barrevie">
                                     <div class="coeur"></div>
                                      <div class="coeur"></div>
                                     <div class="coeur"></div><
                            </div>
                            <div>
                                               <div id="score">score:000</div>
                            </div>
         </div>
                   <div id="buttonbox8">
                            <a href="index.html"><input id="buttonmenu2" type="image" src="img/button(MENU).png"></a>
                   </div>
                   <div id="fond2">
                            <div id="boxtxt2"> </div>
                            <div id="buttonbox4">
                                     <a href="index.html"><input class="buttonmenu" type="image" src="img/button(MENU).png"></a>
                            </div>
                            <div id="buttonbox5">
                                     <a href="game.html"><input class="buttonmenu" type="image" src="img/button(play).png"></a>
                            </div>
                   </div>
         <center><canvas id="monCanvas" width="800px" height="800px"></canvas></center>
</div>
<audio id="player" ></audio>
<script type="text/javascript" src="script.js"></script>
</body>
</html>
```

style.css: @font-face{ font-family: "pixelmania"; src: url(font/Pixelmania.ttf); } @font-face{ font-family: "arcade"; src: url(font/8-bitArcadeIn.ttf); } $.body1\{\\$ background-size: 100%; margin: 0%; padding: 0%; background-image:url("img/fond3(2).png"); background-repeat: no-repeat; } #body2 { background-size: 100%; margin: 0%; padding: 0%; background-image: url("img/fond_jeu.png"); } #buttonbox{ float: right; position: absolute; top: 57%; left:42%; height:2px; width:30px; } .buttonmenu{ margin:15px; width:300px; height:108.5px; } #buttonmenu2{ margin:15px; width:300px; height:108.5px; } #buttonbox2{ position: absolute; float:right; top: 1%;

```
left: 1%;
}
#buttonbox4{
          position: absolute;
          float:right;
          top: 80%;
          left: 15%;
}
#buttonbox8{
          position: absolute;
          float:right;
          top: 70%;
          left: 2%;
}
#buttonbox5{
          position: absolute;
          float:left;
          top: 80%;
          left: 70%;
}
#optionbutton{
          height: 110px;
          width: 110px;
}#optionbutton2{
          float:right;
          margin:60px;
          position: absolute
          height: 50px;
          width: 50px;
}
#gamename{
          float: right;
          position: absolute;
          left:34%;
          top: 4%;
}
.boxtxt{
          margin-top: 1.5%;
          margin-left: auto;
          margin-right: auto;
          width: 90%;
          background-color: rgba(0,0,0,0.8);
          border:10px double;
          border-color: yellow;
```

```
}
#boxtxt2{
          margin-top: 1.5%;
          margin-left: auto;
          margin-right: auto;
          width: 50%;
          background-color: rgba(0,0,0,0.8);
          border:10px double;
          border-color: yellow;
}
#boxbutton3{
          background-color: rgba(0,0,0,0.8);
          display: none;
          margin-left: auto;
          margin-right: auto;
          margin-top: 6%;
          height: 72%;
          width: 35%;
          border:5px solid;
          border-color: yellow;
}
#fond{
          position: absolute;
          height: 100%;
          width: 100%;
          background-color: rgba(0,0,0,0.8);
          display: none;
}
#fond2{
          position: absolute;
          height: 100%;
          width: 100%;
          background-color: rgba(0,0,0,0.8);
          display: none;
}
#boxbutton9{
          background-color: rgba(0,0,0,0.8);
          display: none;
          margin-left: auto;
          margin-right: auto;
          margin-top: 6%;
          height: 72%;
          width: 35%;
          border:5px solid;
          border-color: yellow;
}
```

```
#cross{
          margin: 25px;
          float: right;
          height: 50px;
          width: 50px;
}
#txtregles{
          margin: 25px;
          font-family: 'arcade';
          color: yellow;
          font-size: 40px;
          text-align: justify;
}
#txtregles{
          margin: 25px;
          font-family: 'arcade';
          color: yellow;
          font-size: 40px;
          text-align: justify;
}
#txtregles2{
          margin: 25px;
          font-family: 'arcade';
          color: yellow;
          font-size: 40px;
          text-align: center;
}
#ttrregles{
          color: yellow;
          text-align: center;
          font-family: 'pixelmania';
}
#cadre{
          background-color: rgba(0,0,0,0.5);
          float: left;
          margin:50px;
          height: 300px;
          width: 300px;
          border: double 10px;
          border-color: #0161be;
          position: absolute;
}
```

```
#monCanvas{
margin-top: 20px;
background-image: url(img/labyrinthe.png);
background-size: contain;
background-repeat: no-repeat;
}
#barrevie{
          width: 100%;
          height: 40px;
          top: 0px;
}
.coeur{
          width: 40px;
          height: 40px;
          float: left;
          margin: 12px;
          background-size: contain;
          background-repeat: no-repeat;
          background-image: url(img/vie.png);
}
#bt{
          height: 600px;
          widows: 600px;
          position: absolute;
}
#scoretxt{
          font-family:'arcade';
          color: yellow;
          text-align: center;
          font-size: 50px;
}
```

script2.js:

```
function parametre(){
     var el = document.getElementByld('fond');
     el.style.display= "block";

     var el2 = document.getElementByld('boxbutton3');
     el2.style.display = "block";

}
function parametre2(){
     var el = document.getElementByld('fond');
     el.style.display= "none";

     var el2 = document.getElementByld('boxbutton3');
     el2.style.display = "none";
}
```

script.js: var canvas = document.getElementById('monCanvas'); var ctx = canvas.getContext('2d'); var urlMusique = ["music/vvvvvv.mp3"] var audioEl = document.getElementById("myAudio"); var noQ = 0;var son = false; var mort=false; var mourir=true; var dmrg = true; var jeu = true; var start = false; var vieEnCours =3; var score = 0; console.log("nbr vies :" + vieEnCours); var x=20; var y=10; var niveaux = 1;var directionX=0; var directionY=0; var dx=0; var dy=0; var posx = 1;var posy = 1;var next = ""; var ennemi_img = new Image() ennemi_img.src = "img/squellette.png" var portal1img = new Image() portal1img.src = "img/portail_vert.png" var portal2img = new Image() portal2img.src = "img/portail_rouge.png" var potionimg = new Image() potionimg.src = "img/potion.png"

```
var testimg = new Image()
testimg.src = "img/heosss.png"
var pieceImg = new Image()
pieceImg.src = "img/piece.png"
var piece2Img = new Image()
piece2Img.src = "img/diamond.png"
var piece3Img = new Image()
piece3Img.src = "img/patae.png"
var piece4Img = new Image()
piece4Img.src = "img/item7_3.png"
var piece5Img = new Image()
piece5Img.src = "img/skeleton-head.png"
var swordImg = new Image()
swordImg.src = "img/iron_sword.png"
//Objet pour l'ennemi 0
var ennemi0 = new Object();
ennemi0.directionX = 1;
                            //vecteur de direction en x
ennemi0.directionY = 0;
                                      //vecteur de direction en y
ennemi0.posx = 70;
                                      //position dans le tableau
                                      //position dans le tableau
ennemi0.posy = 65;
En0= true;
//Objet pour l'ennemi 1
var ennemi1 = new Object();
ennemi1.directionX = 1;
ennemi1.directionY = 0;
ennemi1.posx = 70;
ennemi1.posy = 65;
En1= true;
//Objet pour l'ennemi 2
var ennemi2 = new Object();
ennemi2.directionX = 1;
ennemi2.directionY = 0;
ennemi2.posx = 70;
ennemi2.posy = 65;
En2= true;
```

```
//Objet pour l'ennemi 3
var ennemi3 = new Object();
ennemi3.directionX = 1;
ennemi3.directionY = 0;
ennemi3.posx = 70;
ennemi3.posy = 65;
En3= true;
//Objet pour l'ennemi 4
var ennemi4 = new Object();
ennemi4.directionX = 1;
ennemi4.directionY = 0;
ennemi4.posx = 70;
ennemi4.posy = 65;
En4= true;
//Tableau d'ennemis
var ennemis = [];
ennemis.push(ennemi0);
ennemis.push(ennemi1);
ennemis.push(ennemi2);
ennemis.push(ennemi3);
ennemis.push(ennemi4);
//tableau qui dessine le parcours
var deplacements = [
,0,0,0,0],
,0,0,0,0],
,0,0,0,0],
,0,0,0,0],
,0,0,0,0],
```

,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],

,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],

,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], .0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],

[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0
[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0
[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0
[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0
[0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1;
var pieces =[

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

,[0,0,0,0],,0,0,0,0],[0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,7,0,0],

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

,[0,0,0,0],,0,0,0,0],[0,0,0,0][0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],

el.innerHTML = '<h2 id="ttrregles">GAME OVER</h2> votre score est de '+score+' Vous

```
function bbb() {
          if (son == false) {
                    son = true;
                    var el = document.getElementById('optionbutton2')
                    el.src="img/son_actif.png"
          }else if (son == true ) {
                    son =false;
                    var el = document.getElementById('optionbutton2')
                    el.src="img/son_pasactif.png"
          }
          audioEl.src = urlMusique[noQ];
          if (son == false) {
                    audioEl.pause();
          }else if (son == true) {
                    audioEl.play();
          }
}
function parametre(){
          var el = document.getElementById('fond2');
          el.style.display= "block";
}
function scoreFinal() {
```

];

4TTr Dossier de conception 2020-2021 : Leseure Enzo

pouvez recommencer ou retourner au menu principal. '

var el = document.getElementById('boxtxt2')

```
}
if (jeu==true) {
           function finEpee() {
                      mourir=true;
                      testimg.src = "img/heosss.png";
                      ennemi_img.src = "img/squellette.png";
          }
           function random(min, max){
                      var nb = Math.round(Math.random() * (max - min) + min);
          }
           function changeDirection(obstacle, route, perso){
                      var finalDir = "";
                      var nb;
                      //Si obstacle et route sur déplacement horizontal
                      if (obstacle && route && (perso.directionX != 0)){
                      //choisir aléatoirement s'il y a deux routes
                                 if(deplacements[perso.posy+1][perso.posx] == 1 \&\& deplacements[perso.posy+1][perso.posx] == 1) \{ (deplacements[perso.posy+1][perso.posx] == 1) \} \}
                                            nb = random(1,2);
                                            if(nb==1){
                                                        finalDir = "up";
                                            }else{
                                                        finalDir = "down"
                                 //Choisir dessus si route vers dessus
                                 } else if(deplacements[perso.posy-1][perso.posx] == 1){
                                            finalDir = "up";
                                 //Choisir bas si route vers le bas
                                 } else {
                                            finalDir = "down";
                                 }
                      //Si obstacle et route sur déplacement vertical
                      } else if (obstacle && route && (perso.directionY != 0)){
                                 //choisir aléatoirement s'il y a deux routes
                                 if (deplacements[perso.posy][perso.posx-1] == 1 \&\& \ deplacements[perso.posy][perso.posx+1] == 1) \{ (deplacements[perso.posy][perso.posx+1] == 1) \}
                                            nb = random(1,2);
                                            if(nb==1){
                                                        finalDir = "right";
```

```
}else {
                                finalDir = "left"
          //Choisir gauche si route vers gauche
          } else if(deplacements[perso.posy][perso.posx-1] == 1){
                     finalDir = "left";
          //Choisir droite si route vers droite
          } else {
                     finalDir = "right";
          }
//Si route sur déplacement vers la droite
} else if (route && perso.directionX == 1){
           finaldir = "right";
          //S'il y a deux routes annexes choisir aléatoirement entre l'une des deux ou continuer
          if(deplacements[perso.posy+1][perso.posx] == 1 && deplacements[perso.posy+1][perso.posx] == 1){}
                     nb = random(1,3);
                     if(nb==1){
                                finalDir = "right";
                     }else if(nb==2){
                                finalDir = "up"
                     } else {
                                finalDir = "down"
          //Choisir aléatoirement entre continuer et up si route vers dessus
          } else if(deplacements[perso.posy-1][perso.posx] == 1){
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "right";
                     }else if(nb==2){
                                finalDir = "up"
                     }
          //Choisir aléatoirement entre continuer et down si route vers la dessous
          } else {
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "right";
                     }else if(nb==2){
                                finalDir = "down"
                     }
//Si route sur déplacement vers la gauche
} else if (route && perso.directionX == -1){
          finaldir = "left";
          //S'il y a deux routes annexes choisir aléatoirement entre l'une des deux ou continuer
           if(deplacements[perso.posy+1][perso.posx] == 1 \&\& deplacements[perso.posy+1][perso.posx] == 1) \{ if(deplacements[perso.posy+1][perso.posx] == 1) \} \}
```

```
nb = random(1,3);
                     if(nb==1){
                                finalDir = "left";
                     }else if(nb==2){
                                finalDir = "up"
                     } else {
                                finalDir = "down"
                     }
          //Choisir aléatoirement entre continuer et up si route vers dessus
          } else if(deplacements[perso.posy-1][perso.posx] == 1){
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "up";
                     }else if(nb==2){
                                finalDir = "left"
                     }
          //Choisir aléatoirement entre continuer et down si route vers la dessous
          } else {
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "down";
                     }else if(nb==2){
                                finalDir = "left"
                     }
          }
//Si route sur déplacement vers le haut
} else if (route && perso.directionY == -1){
          finaldir = "up";
           //S'il y a deux routes annexes choisir aléatoirement entre l'une des deux ou continuer
           if(deplacements[perso.posy][perso.posx+1] == 1 \&\& \ deplacements[perso.posy][perso.posx+1] == 1) \{ (deplacements[perso.posy][perso.posx+1] == 1) \}
                     nb = random(1,3);
                     if(nb==1){
                                finalDir = "left";
                     }else if(nb==2){
                                finalDir = "right"
                     } else {
                                finalDir = "up"
                     }
           //Choisir aléatoirement entre continuer et left si route vers gauche
          } else if(deplacements[perso.posy][perso.posx-1] == 1){
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "left";
                     }else if(nb==2){
                                finalDir = "up"
                     }
```

```
//Choisir aléatoirement entre continuer et right si route vers la droite
          } else {
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "right";
                     }else if(nb==2){
                                finalDir = "up"
                     }
//Si route sur déplacement vers le haut
} else if (route && perso.directionY == 1){
           finaldir = "down";
           //S'il y a deux routes annexes choisir aléatoirement entre l'une des deux ou continuer
           if (deplacements[perso.posy][perso.posx+1] == 1 \&\& \ deplacements[perso.posy][perso.posx+1] == 1) \{ (deplacements[perso.posy][perso.posx+1] == 1) \}
                     nb = random(1,3);
                     if(nb==1){
                                finalDir = "left";
                     }else if(nb==2){
                                finalDir = "right"
                     } else {
                                finalDir = "down"
                     }
          //Choisir aléatoirement entre continuer et left si route vers gauche
          } else if(deplacements[perso.posy][perso.posx-1] == 1){
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "left";
                     }else if(nb==2){
                                finalDir = "down"
          //Choisir aléatoirement entre continuer et right si route vers la droite
          } else {
                     nb = random(1,2);
                     if(nb==1){
                                finalDir = "right";
                     }else if(nb==2){
                                finalDir = "down"
                     }
          }
}
//Changer la direction en fonction du semi-aléatoire généré
if (finalDir == "up"){
           perso.directionX = 0;
           perso.directionY = -1;
} else if (finalDir == "down") {
```

```
perso.directionX = 0;
                                                        perso.directionY = 1;
                            } else if ( finalDir == "left") {
                                                        perso.directionX = -1;
                                                        perso.directionY = 0;
                            } else if (finalDir == "right"){
                                                        perso.directionX = 1;
                                                        perso.directionY = 0;
                            }
}
 //fonction qui détermine si le perso a un obstacle face à lui
 function obstacle(perso) {
                             var isObstacle = false;
                            if ( perso.directionX == 1 && deplacements[perso.posy][perso.posx+1] == 0
                                                                                   || perso.directionX == -1 && deplacements[perso.posy][perso.posx-1] == 0
                                                                                   ||\ perso.directionY == 1\ \&\&\ deplacements[perso.posy+1][perso.posx] == 0
                                                                                   || perso.directionY == -1 && deplacements[perso.posy-1][perso.posx] ==0
                                                       ) {
                                                        isObstacle = true;
                            }
                             return isObstacle;
}
 //fonction qui détermine si le perso a des routes à coté de lui
 function route(perso) {
      var isroute = false;
      //si le héro se déplace à l'horizontale
      if ( perso.directionX !=0 ) {
                                //vérification croisement
           if(deplacements[perso.posy-1][perso.posx] == 1 \mid \mid deplacements[perso.posy+1][perso.posx] == 1 ) \{ if(deplacements[perso.posy+1][perso.posx] == 1 \} \{ if(deplacements[perso.posy+1][perso.posx] == 1 \} \} \{ if(deplacements[perso.posx] == 1 \} \} \{ if(deplacements[perso.po
                isroute = true;
            }
      //si le héro se déplace à la verticale
      } else if ( perso.directionY != 0) {
           //vérification croisement
           if(deplacements[perso.posy][perso.posx+1] == 1 \mid\mid deplacements[perso.posy][perso.posx+1] == 1) \{ if(deplacements[perso.posy][perso.posx+1] == 1) \}
                isroute = true;
            }
     }
      return isroute;
}
 //Calcule la position du perso
 function positionEnemi(perso){
```

```
var isObst = obstacle(perso);
          var isRoutAnnex = route(perso);
          //Si le perso est dans une impasse il fait demi tour
          if(isObst == true && isRoutAnnex == false){
                    perso.directionX = -perso.directionX;
                    perso.directionY = -perso.directionY;
          //Si un obstacle ou une route --> on pourrait également simplement écrire if(isOst || isRoutAnnex){}
          } else if (isObst == true || isRoutAnnex == true) {
                    changeDirection(isObst, isRoutAnnex,perso);
  }
          //Ensuite,
          //Test pour voir si l'on peut continuer à avancer
          if(perso.directionX != 0){
                    if \ (deplacements[perso.posy+perso.directionY][perso.posx+perso.directionX] == 1) \ \{ \\
                              perso.posx = perso.posx + perso.directionX;
                    }
          } else if (perso.directionY != 0){
                    if \ (deplacements[perso.posy+perso.directionY][perso.posx+perso.directionX] == 1) \ \{ \\
                              perso.posy = perso.posy + perso.directionY;
                    }
          }
}
//fonction de base qui dessine les différents personnages
function drawGeneral(){
          //vider le canvas
          if (start==true) {
                    if (En0==true) {
                              positionEnemi(ennemis[0]);
                              ctx.drawImage(ennemi_img,ennemis[0].posx *10,ennemis[0].posy *10,50,50);
                    }if (En1==true) {
                              positionEnemi(ennemis[1]);
                              ctx.drawlmage(ennemi\_img,ennemis[1].posx~*10,ennemis[1].posy~*10,50,50);\\
                    }if (En2==true) {
                              positionEnemi(ennemis[2]);
                              ctx.drawlmage(ennemi\_img,ennemis[2].posx~*10,ennemis[2].posy~*10,50,50);\\
                    }if (En3==true) {
                              positionEnemi(ennemis[3]);
                              ctx.drawlmage(ennemi\_img,ennemis[3].posx~*10,ennemis[3].posy~*10,50,50);\\
                    }if (En4==true) {
                              positionEnemi(ennemis[4]);
                              ctx.drawlmage(ennemi_img,ennemis[4].posx *10,ennemis[4].posy *10,50,50);
```

```
}
         }
}
//instruction qui lance la fct de dessin toutesles 10 milisecondes
setInterval
                   (draw,60);
function keyDownHandler(e){
         if (e.keyCode == 39) {
                   next = "right";
         }else if(e.keyCode == 37) {
                   next = "left";
         }else if(e.keyCode == 40) {
                   next = "down";
         }else if(e.keyCode == 38) {
                   next = "up";
         }if (dmrg==true) {
                   if(e.keyCode == 32) {
                            start = true;
                            En0=true;
                            En1=true;
                            En2=true;
                            En3=true;
                            En4=true;
                            dmrg==false
                   }
         }
}
function killFantome() {
         if (posy== ennemi0.posy && posx== ennemi0.posx){
                   var dmc_setTimeout = setTimeout(fant0,20000);
                   ennemi0.posx = 70;
                   ennemi0.posy = 65;
                   En0= false;
                   score = score + 100
                   var el=document.getElementById("score")
          el.innerHTML =""+"score:"+score+""
         }else if(posy==ennemi1.posy && posx==ennemi1.posx){
                   var dmc_setTimeout = setTimeout(fant1,20000);
                   ennemi1.posx = 70;
                   ennemi1.posy = 65;
                   En1= false;
```

```
score = score + 100
                 var el=document.getElementById("score")
         el.innerHTML = ""+"score:"+score+""
        }else if(posy==ennemi2.posy && posx==ennemi2.posx){
                 var dmc_setTimeout = setTimeout(fant2,20000);
                 ennemi2.posx = 70;
                 ennemi2.posy = 65;
                 En2= false;
                 score = score + 100
                 var el=document.getElementById("score")
         el.innerHTML = ""+"score:"+score+""
        }else if(posy==ennemi3.posy && posx==ennemi3.posx){
                 var dmc_setTimeout = setTimeout(fant3,20000);
                 ennemi3.posx = 70;
                 ennemi3.posy = 65;
                 En3= false;
                 score = score + 100
                 var el=document.getElementById("score")
         el.innerHTML = ""+"score:"+score+""
        }else if(posy==ennemi4.posy && posx==ennemi4.posx){
                 var dmc_setTimeout = setTimeout(fant4,20000);
                 ennemi4.posx = 70;
                 ennemi4.posy = 65;
                 En4= false;
                 score = score + 100
                 var el=document.getElementById("score")
         el.innerHTML = ""+"score:"+score+""
        }
function fant0(){
         En0=true;
         ennemi0.posx = 70;
         ennemi0.posy = 65;
}
function fant1(){
         En1=true;
         ennemi1.posx = 70;
         ennemi1.posy = 65;
}
function fant2(){
         En2=true;
         ennemi2.posx = 70;
         ennemi2.posy = 65;
function fant3(){
```

```
En3=true;
          ennemi3.posx = 70;
          ennemi3.posy = 65;
}
function fant4(){
          En4=true;
          ennemi4.posx = 70;
          ennemi4.posy = 65;
}
function kill() {
          vieEnCours =vieEnCours - 1;
          start = false
          var el = document.getElementById('barrevie')
          var txt ="";
          var i;
          if (vieEnCours==0) {
                   dmrg=false;
          }else if (vieEnCours>0) {
                    dmrg=true;
          }
          for (i=0;i<vieEnCours;i++) {
                   txt
                             =txt +'<div class="coeur"></div>'
          }
          el.innerHTML = txt;
          console.log(vieEnCours)
}
function vieplus1() {
          vieEnCours =vieEnCours + 1;
          var el = document.getElementById('barrevie')
          var txt ="";
          var i;
          for (i=0;i<vieEnCours;i++) {
                   txt
                             =txt +'<div class="coeur"></div>'
          el.innerHTML = txt;
          console.log(vieEnCours)
}
function draw(){
          if (dmrg==false && vieEnCours==0) {
                    parametre()
```

```
scoreFinal()
}
var rnb;
//Vider le rectangle, dessiner le héro
ctx.clearRect(0,0,canvas.width,canvas.height)
ctx.drawImage(testimg,x,y,50,50)
drawGeneral()
genererPieces();
if (vieEnCours== 0){
         jeu=false;
}
if (mourir==true) {
         if ((posy== ennemi0.posy && posx== ennemi0.posx)
         || (posy==ennemi1.posy && posx==ennemi1.posx)
|| (posy==ennemi2.posy && posx==ennemi2.posx)
|| (posy==ennemi3.posy && posx==ennemi3.posx)
|| (posy==ennemi4.posy && posx==ennemi4.posx)) {
         kill()
                   mort = true;
                   console.log("nbr vies :" + vieEnCours);
         } if (mort==true) {
                   x=20;
                   y=10;
                   posx=1;
                   posy=1;
                   mort=false
                   ennemi0.posx = 70;
                   ennemi0.posy = 65;
                   ennemi1.posx = 70;
                   ennemi1.posy = 65;
                   ennemi2.posx = 70;
                   ennemi2.posy = 65;
                   ennemi3.posx = 70;
                   ennemi3.posy = 65;
                   ennemi4.posx = 70;
                   ennemi4.posy = 65;
                   En0==true;
                   En1==true;
                   En2==true;
                   En3==true;
                   En4==true;
         }
}
if (killFant=true) {
```

```
killFantome()
          }
          if(start==true){
          //si je peux tourner, je change de direction avec la direction 'next'
          // enregistrée par une frappe du clavier
                    if (next == "right" && deplacements[posy][posx+1] == 1) {
                              directionX = 1;
                              directionY = 0;
                    }if ( posx==70 && posy ==31) {
                              directionX = 1;
                              directionY = 0;
                              posx=5;
                    }else if(next == "left" && deplacements[posy][posx-1] == 1) {
                              directionX = -1;
                              directionY = 0;
                    }else if( posx==5 && posy ==31) {
                              directionX = -1;
                              directionY = 0;
                              posx= 70;
                              posy=31;
                    }else if(next == "down" && deplacements[posy+1][posx] == 1) {
                              directionX = 0;
                              directionY = 1;
                    }else if(next == "up" && deplacements[posy-1][posx] ==1) {
                              directionX = 0;
                              directionY = -1;
                    }
                    //Ensuite,
                    //Test pour voir si l'on peut continuer à avancer
                    if(directionX != 0){
                              if (deplacements[posy+directionY][posx+directionX] == 1) {
                                        posx = posx + directionX;
                                        x = (posx)*10;
                              }
                    } else if (directionY != 0){
                              if (deplacements[posy+directionY][posx+directionX] == 1) {
                                        posy = posy + directionY;
                                        y = (posy)*10;
                              }
                    }
                    //pacman mange les pièces
                    mangePiece(posx, posy);
          }
}
```

```
// A chaque dessin du canevas, on génère les pièces sur base du tableau "pièces"
function genererPieces(){
          var i;
          var j;
          var nbPieces = 0;
  for (i=0; i<pieces.length; i++){}
          for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 1){
                     nbPieces ++;
          ctx.drawlmage(piecelmg,j*10+15,i*10+20,20,20);\\
          }
          }
          }for (i=0; i<pieces.length ; i++){</pre>
          for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 6){
          ctx.drawImage(portal1img,j*10-15,i*10-35,75,125);\\
          }
          }
          }for (i=0; i<pieces.length ; i++){</pre>
          for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 7){
          ctx.drawImage(portal2img,j*10-30,i*10-50,75,150);
          }
          }for (i=0; i<pieces.length; i++){
                     for (j=0; j<pieces[0].length; j++){
                     if (pieces[i][j] == 5){
                                nbPieces ++;
                     ctx.drawlmage(swordlmg,j*10+8,i*10+4,40,40);\\
                     }
          } for (i=0; i < pieces.length; i++) {
                     for (j=0; j<pieces[0].length; j++){
                     if (pieces[i][j] == 3){
                                nbPieces ++;
                     ctx.drawImage(potionimg,j*10+8,i*10+4,35,50);\\
                     }
          for (i=0; i< pieces.length; i++){
                     for (j=0; j<pieces[0].length; j++){
                     if (pieces[i][j] == 2){
```

```
nbPieces ++;
          ctx.drawImage(pieceImg,j*10+20,i*10+20,20,20);\\
          }
          }
}if (niveaux==1) {
          for (i=0; i<pieces.length; i++){
          for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 4){
                     nbPieces ++;
           ctx.drawImage(piece2Img,j*10+7,i*10+12,40,40);\\
          }
          }
}
if (niveaux==2) \{
for (i=0; i<pieces.length; i++){
           for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 4){
                     nbPieces ++;
           ctx. drawl mage (piece 3 lmg, j*10+9, i*10+12, 35, 35);\\
          }
          }
if (niveaux==3) \{
          for (i=0; i<pieces.length; i++){
          for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 4){
                     nbPieces ++;
          ctx.drawlmage(piece4lmg,j*10+10,i*10+17,30,30);
          }
          }
}
if (niveaux==4) {
   for (i=0; i<pieces.length; i++){
          for (j=0; j<pieces[0].length; j++){
          if (pieces[i][j] == 4){
                     nbPieces ++;
          ctx.drawImage(piece5Img,j*10+8,i*10+12,50,50);\\
          }
}
```

```
niveaux++;
  console.log("nv"+niveaux)
  pieces =[
[0,0,0,0]
,0,0,0,0],
,0,0,0,0],
[0,0,0,0]
,[0,0,0,0],
,0,0,0,0],
,0,0,0,0],
,[0,0,0,0],
,0,0,0,0],
.0.0.0.01.
,[0,0,0,0],
,0,0,0,0],
,0,0,0,0],
,[0,0,0,0],
```

if (nbPieces == 0 && niveaux==1) {

,[0,0,0,0],,0,0,0,0],[0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],

[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,6,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
[0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0

,0,0,0,0],,0,0,0,0],[0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],

,0,0,0,0],,0,0,0,0],[0,0,0,0][0,0,0,0]]; start=false; x=20; y=10;posx=1; posy=1; mort=false ennemi0.posx = 70; ennemi0.posy = 65; ennemi1.posx = 70; ennemi1.posy = 65; ennemi2.posx = 70; ennemi2.posy = 65; ennemi3.posx = 70; ennemi3.posy = 65; ennemi4.posx = 70; ennemi4.posy = 65; }else if (nbPieces == 0 && niveaux==2) { niveaux++; console.log("nv"+niveaux) pieces =[,0,0,0,0], ,0,0,0,0], ,[0,0,0,0],

,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],

,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,7,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],

,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],

,0,0,0,0], [0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,1,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],,0,0,0,0], ,0,0,0,0],,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],[0,0,0,0],[0,0,0,0],

,0,0,0,0]]; start=false; x=20;y=10; posx=1; posy=1; mort=false ennemi0.posx = 70; ennemi0.posy = 65; ennemi1.posx = 70; ennemi1.posy = 65; ennemi2.posx = 70; ennemi2.posy = 65; ennemi3.posx = 70; ennemi3.posy = 65; ennemi4.posx = 70; ennemi4.posy = 65; }else if (nbPieces == 0 && niveaux==3) { niveaux==1; console.log("nv"+niveaux) pieces =[,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],

,0,0,0,0],.0.0.0.01. ,0,0,0,0],[0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],

,[0,0,0,0],.0.0.0.01. ,0,0,0,0],[0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,7,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],

,0,0,0,0],.0.0.0.01. ,0,0,0,0],[0,0,0,0],0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], ,0,0,0,0], ,[0,0,0,0], .0.0.0.01. ,0,0,0,0],

,0,0,0,0],.0.0.0.01. ,0,0,0,0], [0,0,0,0],0,0,0,0], ,0,0,0,0], ,0,0,0,0], ,0,0,0,0],,0,0,0,0], ,[0,0,0,0],,0,0,0,0], .0,0,0,0]]; start=false; x=20;y=10; posx=1; posy=1; mort=false ennemi0.posx = 70; ennemi0.posy = 65; ennemi1.posx = 70;

```
ennemi1.posy = 65;
                                                     ennemi2.posx = 70;
                                                     ennemi2.posy = 65;
                                                     ennemi3.posx = 70;
                                                     ennemi3.posy = 65;
                                                     ennemi4.posx = 70;
                                                     ennemi4.posy = 65;
                          }
}
//Lorsque que le héro se situe sur une pièce, il la "mange"
 //Cela met à jour le tableau pièces
 function mangePiece(x,y){
                           if(pieces[y][x] == 1){
                           pieces[y][x] = 0;
                           score = score + 10
                           var el=document.getElementById("score")
                           el.innerHTML = ""+"score:"+score+""
                          }else if(pieces[y][x] == 4 && niveaux==1){
                           pieces[y][x] = 0;
                           score = score + 50
                          var el=document.getElementById("score")
                           el.innerHTML = ""+"score:"+score+""
                          }else if(pieces[y][x] == 4 \&\& niveaux==2){
                           pieces[y][x] = 0;
                           score = score + 100
                          var el=document.getElementById("score")
                           el.innerHTML = ""+"score:"+score+""
                          }else if(pieces[y][x] == 4 && niveaux==3){
                           pieces[y][x] = 0;
                           score = score + 200
                           var el=document.getElementById("score")
                           el.innerHTML =""+"score:"+score+""
                          ext{lense} ext{lense
                           pieces[y][x] = 0;
                           score = score + 400
                           var el=document.getElementById("score")
                           el.innerHTML = ""+"score:"+score+""
                          }else if(pieces[y][x] == 5){
                           pieces[y][x] = 0;
                           mourir=false;
```

```
testimg.src = "img/heros.png"
ennemi_img.src = "img/squellettef.png";
var dmc_setTimeout = setTimeout(finEpee,20000);
killFant = true;
}if(pieces[y][x] == 2){
pieces[y][x] = 0;
score = score + 10

var el=document.getElementById("score")
el.innerHTML = ""+"score:"+score+""
}if(pieces[y][x] == 3){
pieces[y][x] = 0;
vieplus1();
}
```

}

table des matières :

Page de garde	Page 1
Definition du projet	Page 2
interfaces	Page 3
Visuel prévu et visuel final	Page 5
Temps de travail	Page 6
Codes	Page 7