Chasse au trésor - Documentation technique

Nicolas CHALOYARD

20 mai 2022

Table des matières

Chapitre 1

Introduction

1.1) Présentation du projet :

Le projet de jeu de Chasse au Trésor est basé sur le double objectif de s'échapper d'une île maudite en ayant trouvé un coffre au trésor tout en survivant.

1.2) Présentation du document :

Ce document est une documentation technique écrite en LaTex sur le projet de chasse au trésor.
Il y sera décrit :

- Les classes ;
- Les attributs ;
- Les méthodes ;

Il y sera également décrit :

- Les soucis techniques ;
- Les manques ;
- Les améliorations é apporter ;
- Les possibles évolutions qui pourraient être apportées ;

Chapitre 2

Partie 1 - Description du fichier jeu.js

2.1) Les classes

Nous décrirons ici les différentes classes présentes dans jeu.js, qu'il s'agisse : des attributs, des méthodes et de leur fonctionnement, etc.

2.1.1) Classe Carte

2.1.1.1 *Description de la classe*

2.1.1.1.1) But de la classe :

La classe **Carte** gère l'affichage de la carte dans le jeu. Cette classe gère également les clicks, en cliquant sur l'une des cases celle-ci ce grise (voir figure 1) De manière encore très rustique il s'agit de case à cliquer :

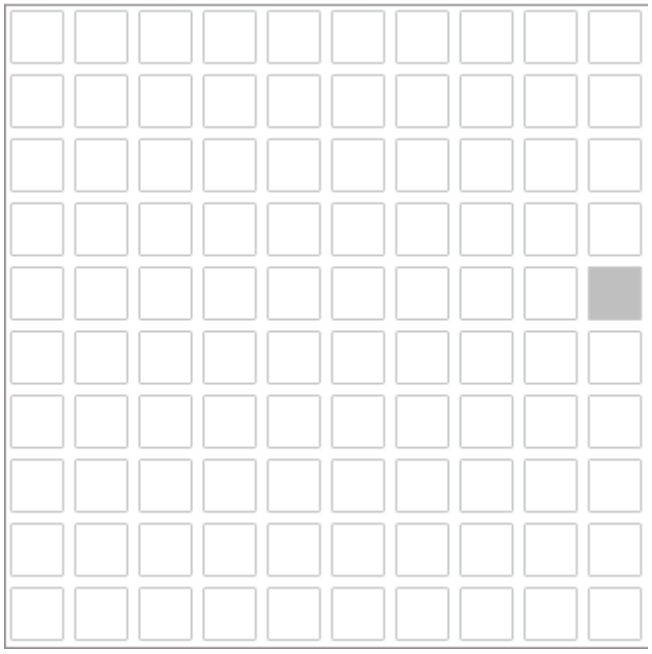

Figure 2.1 – Image de la carte générée

2.1.1.2 *Les attributs*

2.1.1.2.1) Listes des attribut

```
1       // On genere les items sur la carte
2       item = new Item();
3       // On recupere la carte
4       carte = document.getElementById("carte");
5       // Nombre de lignes
6       x = 10;
7       // Nombre de colonnes
8       y = 10;
9       // Tableau des ids des cases selectionnees
10      ids = [];
11
```

Listing 2.1 – Attributs classe Carte

2.1.1.2.2) Description des attributs

- item = Objet de la classe Item permettant l'instanciation des items
- carte = élément du document qui va intégré la carte
- x et y = nombre de lignes et de colonnes permettant la définition de la carte
- ids = tableaux des identifiants des cases sélectionnées

2.1.1.3 *Les méthodes*

2.1.1.3.1) Listes des méthodes

```
1            // Genere la carte
2            initCarte(x, y, carte) {}
3            // Regenere la carte si necessaire
4               regenCarte(x, y, carte) {}
5               // Gere les clicks
6                eventHandler(carte) {}
7            // Gere les clicks sur les noeuds
8            getNodeClicked(event) {}
9            // ajoute les anciens ids des cases cocher dans le tableau "
    id"
10              ajoutAncienID(id) {}
11              // Affiche tout les ids stockes
12              getAllID()
13              // Renvoie l'ID de l'index choisi
14              getID(index)
15           // Genere les items sur la carte
16           genItem()
17
```

Listing 2.2 – Méthodes classe Carte

2.1.1.3.2) Description des fonctions

- initCarte(x, y, carte) = initialise la carte ainsi que les cases clickable,en fonction d'un taille x et y et d'un emplacement de carte
- regenCarte(x, y, carte) = réinitialise la carte avec les mêmes paramètres que initcarte
- eventHandler(carte) = gère les événements de la carte (les clicks), retourne vrai si l'élément cliqué correspond à une case faux sinon
- getNodeClicked(event) = gère quels noeuds ont été cliqués en fonction d'un événement (définit dans eventHandler)
- ajoutAncienID(id) = ajoute l'id cliqué dans le tableau des ids
- getAllID() = affiche tout les items déjà cliqués
- getID(index) = retourne l'id de l'index voulu
- genItem() = génère les items dans la carte, il peut s'agir de malus ou de bonus, fonction pas implémenté à l'heure actuelle.

2.1.2) Classe Message

2.1.2.1 *Description de la classe*

2.1.2.1.1) But de la classe :

La classe **Message** gère l'affichage des messages dans le jeu. Cette classe ne gère que l'ajout des messages au cours du jeu. Cependant le jeu n'étant pas fini, les messages ne sont pas implémentés.

2.1.2.2 *Les attributs*

2.1.2.2.1) Listes des attribut

```
1       // On recupere l'emplacement des messages
2       message = document.getElementById("message");
3
```

Listing 2.3 – Attributs classe Message

2.1.2.2.2) Description des attributs

— message = élément du document qui va intégré les messages

2.1.2.3 *Les méthodes*

2.1.2.3.1) Listes des méthodes

```
1      // Genere la zone des messages
2         initMessage() {}
3      // Ajoute le contenu du message
4      ajoutMessage(contenu)
5
```

Listing 2.4 – Méthodes classe Message

2.1.2.3.2) Description des fonctions

— initMessage() = initialise la zone des messages
— ajoutMessage(contenu) = ajoute le message à la zone de message

2.1.3) Classe Item

2.1.3.1 *Description de la classe*

2.1.3.1.1) But de la classe :

La classe **Jeu** gère les items dans le jeu, c'est la classe mère des classes Objet et Malus. Cependant le jeu n'étant pas fini, les Items ne sont pas implémentés.

2.1.3.2 *Les attributs*

2.1.3.2.1) Listes des attribut

```
1     // On recupere l'emplacement des messages
2     Zmessage = document.getElementById("message");
3     // infos des items
4     info = [];
5     // Enumeration des effets possibles
6     effet = { MALUS: 1, BONUS: 0 };
7
8
```

Listing 2.5 – Attributs classe Message

2.1.3.2.2) Description des attributs

- Zmessage = élément du document qui va intégré les messages
- info = tableaux contenant les informations des items
- effet = énumérations des possibilités qu'il s'agissent d'un malus ou d'un bonus

2.1.3.3 *Les méthodes*

2.1.3.3.1) Listes des méthodes

```
1     // Retourne le nom de l'item
2     getNom() {}
3     // Met le nom d'un item a jour
4     setNom(nom)
5     // Retourne l'effet de l'item (malus ou bonus)
6     getEffet() {}
7     // Met a jour effet
8     setEffet(effet) {}
9     // Prototype de fonction permettant la generation d'un nb aleatoire
      d'item (utile pour les sous classes)
10    nbAleat(value) {}
11    // Ajoute l'item dans la liste info
12    ajoutItem(obj)
13    // Retourne le nombre d'item present la liste info
14    getNbItem()
15    // Renvoie les infos d'un objet
16    getInfosObjets()
17    // ajout d'un message dans la zone de message
18    addMessage(obj)
19
```

Listing 2.6 – Méthodes classe Message

2.1.3.3.2) Description des fonctions

- getNom(), setNom() = getter / setter du nom de l'objet
- getEffet(), setEffet() = getter / setter de l'effet objet
- ajoutMessage(contenu) = ajoute le message à la zone de message
- nbAleat(value) = génère un nombre d'item aléatoire entre 1 et value
- ajoutItem(obj) = ajoute l'item dans la liste info
- getNbItem() = renvoie le nombre d'item présent dans la liste info
- getInfosObjets() = renvoie les infos d'un item dans la liste info
- addMessage(obj) = ajoute un message dans la zone de message