

# Documentation technique

## L'essentiel

Application développée en modèle 3 tiers (GUI, Métier (BLL), ~~Accès-BDD~~, Objets (BO)).

Couverture des tests avec AxoCover

Les objets « Aventurier », « Trésor » et « Montagne » héritent de l'interface IObjetCarte dans un souci de maintenabilité car ils pourraient évoluer et avoir de nouveaux points communs.

## Points importants

- Les aventuriers se déplacent chacun leur tour
- Une collision annule le déplacement
- Ne pas sortir de la carte (les positions négatives sont à proscrire)

## Détails de fonctions

Carte.Clone() :

Lorsque l'on passe l'objet « Carte » à traiter pour faire avancer les personnages, il est possible de la traiter directement.

Cependant, créer un clone de celle-ci permet d'obtenir une version traitée de la carte et conserver l'ancienne version. Utile si une option de comparaison est requise un jour.

```
List<string> ReadSortie(Carte carte)
```

Génère une liste de lignes correspondantes au format de sortie d'une carte tel qu'il est demandé en exemple

```
List<string> ReadCarte(Carte carte)
```

Génère une liste de lignes correspondantes au format d'entrée d'une carte. Permet de la visualiser dans l'interface.

---

```
List<string>ReadFile (string sChemin)
```

Récupère le chemin d'un fichier texte et le convertit en une liste de string se basant sur les lignes inscrites dans le fichier.

Cette fonction n'a pas de test unitaire car je ne maîtrise pas le Mock en C# pour passer outre la lecture du fichier

---

```
Carte ChargerCarte(List<string> lines)
```

Créer un objet « Carte » avec ses montagnes, trésors et aventuriers avant traitement à partir d'un ensemble de lignes.

Les collisions sont gérées

Il ne peut pas y avoir 2 lignes « C »

---

```
Carte Traitement(Carte entree)
```

Créer un objet « Carte » sortie qui contient les informations après le traitement de la carte au trésor  
➔ Les aventuriers se sont déplacés et ont ramassé des trésors.

---

```
public bool CheckCollision(List<Position> positions, Position positionATester)
```

Contrôle si l'objet en cours de déplacement ou de création ne sera pas sur la même case qu'un autre présent dans une liste.

On ne passe pas en paramètre la carte mais car on filtre en amont les objets à contrôler. Un trésor et un aventurier peuvent entrer en collision mais pas une montagne et un aventurier.

---

```
public bool CaracteresAutorises(char[] arrayAutorise, char[] arrayTest )
```

Contrôle si les actions et les orientations indiquées sur la carte existent dans le système