Indique ici la collection concernée dans le projet de jeu du pendu : (Exemple : PartiePendu.dejaJouees)

Mot.structure = la structure du mot à deviner lors d'une partie de pendu.

Critères	Valeurs	Attendus
Principaux besoins	– Retrouver facilement les positions d'une lettre dans un mot – Accéder aux positions sur base d'une clé : la lettre – Les positions de ces lettres sont stockées dans une collection. – Il n'y a pas besoin de trier les clés.	Liste ici les principaux besoins : sur quelle base doit-on accéder à un élément ? un indice ? une clé ? Doivent-ils être triés ? …
Interface choisie	Map<Character, List<Integer>>	Indique l'interface qui te parait la plus pertinente
Justifie !	La Map permet d'associer une clé (ici une lettre) à une valeur (ici une collection des positions auxquelles la lettre apparait dans le mot à deviner). (On pourrait aussi justifier le choix de la liste pour les positions !)	Explique pourquoi cette interface peut convenir, en faisant le lien avec l'énoncé et le résultat à obtenir
Principales méthodes utilisées de cette interface	– get pour récupérer la liste sur base de la lettre – containsKey pour vérifier si une lettre est dans le mot – put pour remplir la map avec chaque lettre du mot à deviner et ses positions dans le mot	Quelles méthodes de cette interface te paraissent intéressantes pour obtenir le résultat souhaité ? Exemple : List.get(int) parce qu'elle permet un accès sur base du numéro d'ordre
Implémentation choisie	HashMap	Indique l'implémentation la plus adéquate pour l'interface choisie plus haut
Justifie !	Les clés (les lettres du mot) n'ont pas besoin d'être triées => une TreeMap ne se justifie pas. Toutes les méthodes utilisées le sont en temps constant (en O(1)). Le type de la clé (Character) fournit une méthode hashCode efficace. La TreeMap (l'autre implémentation de l'interface Map) fournirait des méthodes en O(log(N)), avec N = le nombre d'éléments, ce qui serait moins efficace.	Explique pourquoi cette implémentation peut convenir, en faisant le lien avec l'énoncé et le résultat à obtenir et **en mentionnant les complexités temporelles des méthodes utilisées**