

Exercice 1.

Voici un backlog supplémentaire pour ceux qui ont épuisé le précédent.

- certaines boites ne peuvent être ouvertes qu’avec un certain objet (une clé particulière, une carte magnétique, etc. . .). `b.set_key(t)` enregistre pour la boite `b` le fait qu’elle ne peut être ouverte qu’avec l’objet `t`. Après cela, si `b` est fermée, alors elle ne peut être ouverte qu’avec la nouvelle méthode `b.open_with(t)` avec le bon paramètre. `b.open()` n’a plus l’effet d’ouvrir la boite.
- une boite est une chose. attention : ceci est un refactoring. son but est de permettre de mettre des boites dans des boites (voir item suivant).
- on peut mettre une boite dans une boite.
- un joueur (`Player`) est une chose. il a un nom. et on peut le mettre dans une boite.
- un joueur à une boite (son inventaire) de capacité limitée.
- un lieu (`Location`) est une boite de capacité illimitée.
- toute chose à un attribut `container` qui est la boite dans laquelle il se trouve, ou `None` s’il n’est pas situé dans une boite. cet attribut doit être mis à jour quand on ajoute ou enlève une chose d’une boite.
- soit un joueur `j` : `j.take("bidule")` cherche l’objet nommé "bidule" dans le `container` du joueur (c’est à dire le lieu dans lequel il se trouve), le prend et le met dans son inventaire. `take` renvoie un booléen indiquant si l’action a réussi ou échoué.
- `j.drop("bidule")` prend l’objet nommé "bidule" dans son inventaire et le met dans son `container`.
- `j.open("coffre")` cherche l’objet nommé "coffre" dans l’inventaire ou le `container` du joueur, et l’ouvre.
- `j.close("coffre")` cherche l’objet nommé "coffre" dans l’inventaire ou le `container` du joueur, et le ferme.
- `j.open_with("coffre", "passe-partout")` cherche l’objet nommé "coffre" dans l’inventaire ou le `container` du joueur, et l’objet nommé "passe-partout" juste dans l’inventaire, et applique la méthode `open_with` au premier, avec le second en paramètre.