

# Rapport de test du Usecase Drink

## Cas d'utilisation:

L'application doit permettre à un utilisateur de boire un liquide.

## Specification

Exigences	Description
Ex D1	Le système doit vérifier que l'utilisateur possède le liquide.
Ex D2	Le niveau d'intoxication de l'utilisateur varie selon le liquide bu.
Ex D3	La barre d'ivresse du joueur a augmenter
Ex D4	Le système doit enlever le liquide dans l'inventaire de l'utilisateur après avoir été bu.
Ex D5	Le niveau de vie de l'utilisateur varie selon le liquide bu.

## Environnement de Test

- **Système d'exploitation:** Mac OS 14.5
- **Version de Java:** JDK 17
- **Outils:** JUnit 5.7.1, Gradle 8.7

## Configuration 1 :

### Objectif du test :

Vérifie que le joueur ne peut pas boire de liquide si aucun liquide est dans son inventaire.

### Précondition

- Le joueur ne possède pas de liquide dans son inventaire

## Étapes du Déroulement du Test

**Étape 1 :** Initialiser l'état du joueur sans liquide dans l'inventaire.

**Étape 2 :** Faire boire le joueur avec un indice qui correspond a case vide de l'inventaire

**Étape 3 :** Le système vérifie que la fonction drink renvoie false .

### Postcondition

- La fonction "drink" doit renvoyer false.

### Constater que

- Le joueur n'a consommé a consommé aucun liquide

## Exigences

- **D1** :Le système doit vérifier que l'utilisateur possède le liquide.

## Résultats des Tests

- **Cas de Test 1:** Vérifie que le joueur ne pas boire de liquide si aucun liquide est dans son inventaire
- **Constat:** Le joueur n'a bu aucun liquide
- **Résultat:** Succès.

## Conclusion

Le test vérifie avec succès que le joueur ne peut pas boire de liquide si aucun liquide est dans son inventaire.

---

## Configuration 2 :

### Objectif du test :

Vérifier que le joueur peut boire un liquide.

### Précondition

- Le joueur possède du liquide dans son inventaire

## Étapes du Déroulement du Test

**Étape 1** : Ajouter du liquide dans l'inventaire du joueur.

**Étape 2** : Le joueur boit le liquide contenu dans son inventaire.

### Postcondition

La fonction "drink" renvoie true.

### Constater que

- Le joueur a bu le liquide

## Exigences

- **D1** : Le système doit vérifier que l'utilisateur possède le liquide.
- **D3** : L'utilisateur a la possibilité de boire un liquide.

## Résultats des Tests

- **Cas de Test 2:** Le joueur boit le liquide.
- **Constat:** Le joueur a bu le liquide.
- **Résultat:** Succès.

## Conclusion

Le test vérifie avec succès que le joueur peut boire un liquide.

---

## Configuration 3 :

### Objectif du test :

Vérifie que le joueur a récupère de la vie après avoir bu un liquide

### Précondition

- Le joueur possède une potion de vie dans son inventaire
- Le joueur possède la moitié de son niveau de vie maximum.

## Étapes du Déroulement du Test

**Étape 1 :** Initialiser l'état du joueur avec une potion de vie dans l'inventaire.

**Étape 2 :** Enregistrer le niveau de vie du joueur.

**Étape 3 :** Le joueur boit le liquide dans son inventaire.

**Étape 4 :** Comparer l'ancien niveau de vie et le nouveau.

### Postcondition

- `oldNiveauVie < newNiveauVie`
- `Drink` renvoie `true`

### Constater que

- Le niveaux de vie du joueur a augmenter
- Le joueur a bu le liquide

## Exigences

- **D3 :** L'utilisateur a la possibilité de boire un liquide.
- **D5 :** Le niveau de vie de l'utilisateur augmente adaptativement après avoir bu un liquide.

## Résultats des Tests

- **Cas de Test 3:** Vérifie si le joueur a récupère de la vie après avoir bu un liquide
- **Constat:** Le niveaux de vie du joueur a augmenter
- **Résultat:** Succès.

## Conclusion

Le test vérifie avec succès que joueur a récupère de la vie après avoir bu un liquide

---

## Configuration 4 :

### Objectif du test :

Vérifie que la barre d'ivresse s'adapte après bu certains type de liquide

### Précondition

- Le joueur possède du rhum
- Le niveau d'ivresse du joueur est a zéro

## Étapes du Déroulement du Test

**Étape 1** : Initialiser l'état du joueur avec du rhum dans l'inventaire.

**Étape 2** : Enregistrer la barre d'ivresse du joueur.

**Étape 3** : Le joueur boit le liquide dans son inventaire

**Étape 4** : Comparer l'ancien niveau d'ivresse et le nouveau.

### Postcondition

- `oldNiveauIvresse < newNiveauIvresse`
- `Drink` renvoie `true`

### Constater que

- La barre d'ivresse du joueur a augmenter
- Le joueur a bu le rhum

## Exigences

- **D1** :Le système doit vérifier que l'utilisateur possède le liquide.
- **D3** : L'utilisateur a la possibilité de boire un liquide.
- **D2** :Le niveau d'intoxication de l'utilisateur varie selon le type de liquide bue .

## Résultats des Tests

- **Cas de Test 4:** Vérifie que la barre d'ivresse s'adapte après bu certains type de liquide
- **Constat:** La barre d'ivresse du joueur a augmenter.
- **Résultat:** Succès.

## Conclusion

Le test vérifie avec succès que la barre d'ivresse s'adapte après bu certains type de liquide

---

## Configuration 5 :

### Objectif du test :

Vérifier que le joueur ne peut boire que des liquides.

### Précondition

- Le joueur ne possède pas de liquide dans son inventaire
- Le joueur possède des item

## Étapes du Déroulement du Test

**Étape 1 :** Ajouter une arme dans l'inventaire du joueur

**Étape 2 :** Le testeur fait boire l'arme au joueur

### Postcondition

- Drink renvoie false

### Constater que

- Le joueur n'a rien consommé

### Exigences

- **D1 :** Le système doit vérifier que l'utilisateur possède le liquide.

## Résultats des Tests

- **Cas de Test 5:** Vérifie que le joueur ne peut pas boire que du liquide
- **Constat:** Le joueur n'a pas bu l'item
- **Résultat:** Succès.

### Conclusion

Le test vérifie avec succès que le joueur ne peut pas boire que du liquide

---

## Configuration 6 :

### Objectif du test :

Vérifie que le joueur ne possède plus l'item consommé dans son inventaire

### Précondition

- Le joueur possède un liquide dans son inventaire

## Étapes du Déroulement du Test

**Étape 1** : Initialiser l'inventaire du joueur avec une potion de vie dans son inventaire

**Étape 2** : faire boire le liquide au joueur

**Étape 3** : Vérifier que la potion de vie n'est plus dans l'inventaire.

## Postcondition

- drink renvoie true
- Inventaire du joueur vide.

## Constater que

- Le joueur n'a plus la potion de vie dans son inventaire.

## Exigences

- **D1** :Le système doit vérifier que l'utilisateur possède le liquide.
- **D4** : Le système doit enlever le liquide dans l'inventaire de l'utilisateur après avoir été bu.

## Résultats des Tests

- **Cas de Test 6:** Vérifier que le joueur ne possède plus l'item consommé dans son inventaire
- **Constat:** La potion de vie n'est plus dans l'inventaire après consommation.
- **Résultat:** Succès.

## Conclusion

Le test vérifie avec succès que le joueur n'a plus le liquide consommé dans son inventaire après l'avoir bu.

## Conclusion Final

Ce rapport de test montre que chaque test a été effectué avec succès selon les configurations spécifiées et les exigences correspondantes. Le système répond correctement aux comportements attendus pour le use case Drink, garantissant une fonctionnalité fiable et conforme aux spécifications.