|  |
| --- |
|  |
| **Dossier de Tests** |
| Projet : Consignes et assistance vidéolink |

|  |
| --- |
| Damien GUÉDON  BTS2 |

**Sommaire**

[I. Présentation de la fonction 2](#_Toc421656440)

[II. Tests de la fonction 3](#_Toc421656441)

[**Test n°1** 3](#_Toc421656442)

[**Test n°2** 6](#_Toc421656443)

[**Test n°3** 8](#_Toc421656444)

[**Test n°4** 10](#_Toc421656445)

[**Test n°5** 12](#_Toc421656446)

**TESTS DE LA FONCTION "createBoxes "**

1. **Présentation de la fonction**

**Classe :** BoxChoice

**Fonction :** void createBoxes (const QList<struct\_DoorGeom>&) .

**Rôle de la fonction :**

Elle permet de créer des consignes en fonction de leur taille et d’afficher leurs états (disponible ou indisponible) graphiquement. L’utilisateur aura un visuel pour sélectionner les consignes qu’il souhaite.

**Principe de la fonction**:

La fonction dessine les consignes en fonction de la liste reçue en paramètre et les affiches dans un QGraphicView.

**Paramètre :** QList<struct\_DoorGeom>&

La fonction prend en paramètre une QList<struct\_DoorGeom>. Cette liste est récupérée quand on appelle la fonction GetBoxesList() de la classe CSQLite\_Local\_DB.

**#include** "DataStructures.h"

struct struct\_DoorGeom

{

int BoxNumber;

QString BoxSize;

int status;

int x;

int y;

};

La structure *struct\_DoorGeom* comporte **BoxNumber** le numéro de la consigne, **BoxSize** le taille ("A", "AW", "CM" ou "CW"), **status** l’état de la consigne (0 : OK, 1 : KO, 2 : SELECTED), **x** et **y** sont les coordonnées de la position de la consigne.

**Retours :**

La fonction affiche les consignes créées avec leurs états (couleur vert : disponible (OK), couleur rouge : indisponible (KO)) et le numéro de la consigne.

***Taille de la consigne* :**

* Si **BoxSize** vaut "A" ou "AW", la taille est de 100x100.
* Si **BoxSize** vaut "CM" ou "CW", la taille est de 100x200.

Si la liste est vide, elle affiche, alors, un message d’erreur :" Aucune consigne trouvée. Merci de contacter la maintenance. ".

Si la liste comporte des tailles de consignes invalides (différent de "A", "AW", "CM" ou "CW"), la fonction créera ces consignes avec une taille par défaut (100x100) et les affichera en indiquant que la taille est inconnue.

Si la liste comporte des états de consignes inconnues (différent de OK : 0, KO : 1, SELECTED : 2), la fonction créera ces consignes et les affichera avec un fond blanc pour l’état.

1. **Tests de la fonction**

**Procédure/principe de test :**

***Principe de test :***

Vérifier la création des consignes (taille et état).

***Moyens :***

Pour faire le test, il faut :

* être sous Linux.
* avoir l’exécutable "createboxestest". Cette exécutable a été créée sur QT à l’aide de la librairie QTestLib.

**🗸** **Test n°1**

**Objectif du test :**

Le but de ce test est de prouver que la fonction crée et affiche les consignes avec une taille et un état valide quand elle reçoit en paramètre une liste normale.

**Procédure du test :**

Pour ce test, le programme se charge de créer 4 consignes et d’assigner à chacune un état (libre ou occupée) ainsi qu’une taille valide.

- étape 1 : Lancer le programme "./createboxestest test\_createBoxe\_UseNormal " dans un terminal.

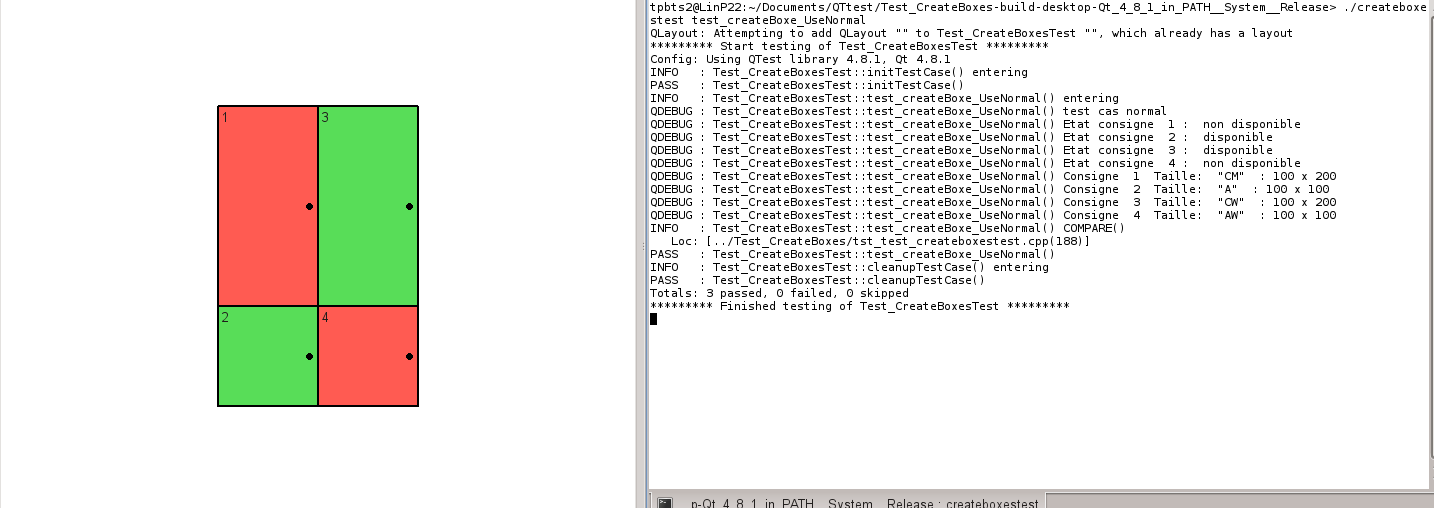
- étape 2 : Vérifier l’affichage des 4 consignes.

- étape 3 : Vérifier les états des consignes (couleur vert : disponible, couleur rouge : indisponible).

- étape 4 : Vérifier les tailles des consignes ("A" ou "AW"(100x100), "CM" ou "CW"(100x200)).

**Résultats attendus :**

* affichage des 4 consignes : OK
* affichage des états (couleur vert : disponible, couleur rouge : indisponible) : OK
* les tailles sont correct : OK



**Résultats obtenus :**

* affichage des 4 consignes : OK
* affichage des états (couleur vert : disponible, couleur rouge : indisponible) : OK
* les tailles sont correct : OK

**Historique des problèmes :**

Aucun problème.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Projet :** Locker\_Control | | **Auteur :** Guédon Damien | | | **Date :** 26/05/2015 |
| **Référence du test :** BoxChoice\_CreateBox\_V2 | | | **Nom du module :** createBoxes | | |
| **Objectifs du test :** Le but de ce test est de prouver que la fonction crée et affiche les consignes avec une taille et un état valide quand elle reçoit en paramètre une liste normale. | | | | | |
| **Conditions du test** | | | | | |
| **Logiciel :**  -Application test : createboxestest  -OS : Linux. | | | | | |
| **Procédure de test** | | | | | |
| **Opérations** | Pour ce test, le programme se charge de créer 4 consignes et assigner à chacune un état (libre ou occupée) ainsi qu’une taille valide.  - étape 1 : Lancer le programme en tapant la commande  "./createboxestest test\_createBoxe\_UseNormal" dans un terminal.  - étape 2 : Vérifier l’affichage des 4 consignes.    - étape 3 : Vérifier les états des consignes (couleur vert : disponible, couleur rouge : indisponible) en comparant les valeurs affichées dans le terminal.  - étape 4 : Vérifier les tailles des consignes ("A" ou "AW"(100x100), "CM" ou "CW"(100x200)) en comparant les valeurs affichées dans le terminal. | | | | |
| **Résultats attendus** | | | | **Résultats obtenus** | |
| * affichage des 4 consignes : OK * affichage des états (couleur vert : disponible, couleur rouge : indisponible) : OK * les tailles sont correctes : OK | | | | * OK * OK * OK | |
| **Conclusion :** OK | | | | | |

**🗸** **Test n°2**

**Objectif du test :**

Le but de ce test est de prouver que la fonction affiche un message d’erreur si la liste passée en paramètre est vide.

**Procédure du test :**

Pour ce test, le programme se charge de mettre une liste vide en paramètre.

- étape 1 : Lancer le programme

"./createboxestest test\_createBoxe\_UseListEmpty " dans un terminal.

- étape 2 : Vérifier si le programme n’affiche aucune consigne mais à la place affiche un message "Aucune consigne trouvée. Merci de contacter la maintenance. "

**Résultats attendus :**

* affichage : vide (aucune consigne créée) : OK
* affichage du message "Aucune consigne trouvée. Merci de contacter la maintenance. " : OK



**Résultats obtenus :**

* affichage : vide (aucune consigne créée) : OK
* affichage du message "Aucune consigne trouvée. Merci de contacter la maintenance. " : OK

**Historique des problèmes :**

* problème d’encodage sur le message erreur.

⮱ Solution : convertir le message d’erreur en utf8.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Projet :** Locker\_Control | | **Auteur :** Guédon Damien | | | **Date :** 26/05/2015 |
| **Référence du test :** BoxChoice\_CreateBox\_V2 | | | **Nom du module :** createBoxes | | |
| **Objectifs du test :** Le but de ce test est de prouver que la fonction affiche un message d’erreur si la liste passée en paramètre est vide. | | | | | |
| **Conditions du test** | | | | | |
| **Logiciel :**  -Application test : createboxestest  -OS : Linux | | | | | |
| **Procédure de test** | | | | | |
| **Opérations** | Pour ce test, le programme se charge de mettre une liste vide en paramètre.  - étape 1 : Lancer le programme en tapant la commande  "./createboxestest test\_createBoxe\_UseListEmpty" dans un terminal.  - étape 2 : Vérifier si le programme n’affiche aucune consigne mais à la place affiche un message "Aucune consigne trouvée. Merci de contacter la maintenance. ". | | | | |
| **Résultats attendus** | | | | **Résultats obtenus** | |
| * affichage : vide (aucune consigne créée) : OK * affichage du message "Aucune consigne trouvée. Merci de contacter la maintenance. " : OK | | | | * OK * OK | |
| **Conclusion :** OK | | | | | |

**🗸** **Test n°3**

**Objectif du test :**

Le but de ce test est de prouver que la fonction affiche un message d’erreur sur chaque consigne qui a une taille invalide. Si la taille est invalide, la fonction doit afficher la consigne avec une taille par défaut (100x100).

**Procédure du test :**

Pour ce test, le programme se charge de créer et mettre une liste de 4 consignes avec des tailles invalides (sauf sur la 1ère consigne) en paramètre.

- étape 1 : Lancer le programme

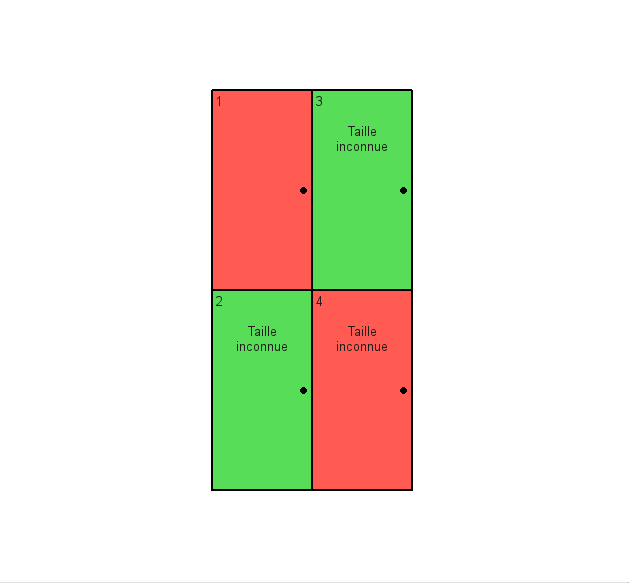
"./createboxestest test\_createBoxe\_UseTypeBoxInvalid " dans un terminal.

- étape 2 : Vérifier à l’affichage la taille des consignes (tailles "A" ou "AW " ou par défaut: 100x100 sinon tailles "CM" et "CW" : 100x200) en comparant les valeurs affichées dans le terminal.

- étape 3 : Vérifier que sur chaque consigne qui a une taille invalide, il y a un message d’erreur du type "Taille inconnue ".

**Résultats attendus :**

* Taille de consigne invalide : affichage de la consigne avec une taille par défaut (taille : 100x100) : OK.
* message erreur sur la consigne "Taille inconnue" : OK.



**Résultats obtenus :**

* taille de consigne invalide : affichage de la consigne avec une taille par défaut (taille : 100x100) : OK
* message erreur sur la consigne "Taille inconnue " : OK

**Historique des problèmes :**

Aucun Problème.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Projet :** Locker\_Control | | **Auteur :** Guédon Damien | | | **Date :** 26/05/2015 |
| **Référence du test :** BoxChoice\_CreateBox\_V2 | | | **Nom du module :** createBoxes | | |
| **Objectifs du test :** Le but de ce test est de prouver que la fonction affiche un message d’erreur sur chaque consigne qui a une taille invalide. Si la taille est invalide, la fonction doit afficher la consigne avec une taille par défaut (100x100). | | | | | |
| **Conditions du test** | | | | | |
| **Logiciel :**  -Application test : createboxestest  -OS : Linux | | | | | |
| **Procédure de test** | | | | | |
| **Opérations** | Pour ce test, le programme se charge de créer et de mettre une liste de 4 consignes avec des tailles invalides (sauf sur la 1ère consigne) en paramètre.  - étape 1 : Lancer le programme en tapant la commande  "./createboxestest test\_createBoxe\_UseTypeBoxInvalid " dans un terminal.  - étape 2 : Vérifier à l’affichage la taille des consignes (tailles " A" ou " AW" ou par défaut : 100x100 sinon tailles " CM" ou " "CW" : 100x200) en comparant les valeurs affichées dans le terminal.  - étape 3 : Vérifier que sur chaque consigne qui a une taille invalide, il y a un message d’erreur du type "Taille inconnue". | | | | |
| **Résultats attendus** | | | | **Résultats obtenus** | |
| * taille de consigne invalide : affichage de la consigne avec une taille par défaut (taille : 100x100) : OK * message erreur sur la consigne "Taille inconnue" : OK | | | | * OK * OK | |
| **Conclusion :** OK | | | | | |

**🗸** **Test n°4**

**Objectif du test :**

Le but de ce test est de prouver que la fonction affiche un fond blanc sur chaque consigne qui possède un état invalide.

**Procédure du test :**

Pour ce test, le programme se charge de créer et mettre une liste de 4 consignes avec des états invalides (sauf sur la 1ère consigne) en paramètre.

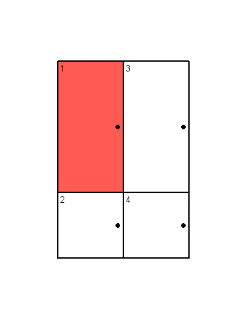
- étape 1 : Lancer le programme en tapant la commande

"./createboxestest test\_createBoxe\_UseStatusBoxInvalid" dans un terminal.

- étape 2 : Vérifier à l’affichage l’état des consignes (couleur vert : disponible, couleur rouge : indisponible et couleur blanc : état inconnu).

**Résultats attendus :**

* état de consigne invalide : affichage fond blanc dans la consigne : OK



**Résultats obtenus :**

* état de consigne invalide : affichage fond blanc dans la consigne : OK

**Historique des problèmes :**

Aucun Problème.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Projet :** Locker\_Control | | **Auteur :** Guédon Damien | | | **Date :** 26/05/2015 |
| **Référence du test :** BoxChoice\_CreateBox\_V2 | | | **Nom du module :** createBoxes | | |
| **Objectifs du test :** Le but de ce test est de prouver que la fonction affiche un fond blanc sur chaque consigne qui possède un état inconnu. | | | | | |
| **Conditions du test** | | | | | |
| **Logiciel :**  -Application test : createboxestest  -OS : Linux | | | | | |
| **Procédure de test** | | | | | |
| **Opérations** | Pour ce test, le programme se charge de créer et mettre une liste de 4 consignes avec des états inconnus (sauf sur la 1ère consigne) en paramètre.  - étape 1 : Lancer le programme en tapant la commande  "./createboxestest test\_createBoxe\_UseStatusBoxInvalid" dans un terminal.  - étape 2 : Vérifier à l’affichage l’état des consignes (couleur vert : disponible, couleur rouge : indisponible et couleur blanc : état inconnu) en comparant les valeurs affichées dans le terminal. | | | | |
| **Résultats attendus** | | | | **Résultats obtenus** | |
| * état inconnu d’une consigne : affichage fond blanc dans la consigne : OK | | | | * OK | |
| **Conclusion :** OK | | | | | |

**🗸** **Test n°5**

**Objectif du test :**

Le but de ce test est de prouver que lorsqu’on clique sur une consigne la couleur change.

**Procédure du test :**

Pour ce test, le programme se charge de créer 4 consignes et d’assigner à chacune un état (libre ou occupée) ainsi qu’une taille valide.

* étape 1 : Lancer le programme en tapant la commande

"./createboxestest test\_createBoxe\_UseStatusBoxInvalid " dans un terminal.

* étape 2 : Vérifier que lorsqu’on clique sur une consigne celle-ci change de couleur.

**Résultats attendus :**

* changement de couleur d’une consigne : OK

**Résultats obtenus :**

* changement de couleur d’une consigne : OK

**Historique des problèmes :**

Aucun Problème.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Projet :** Locker\_Control | | **Auteur :** Guédon Damien | | | **Date :** 26/05/2015 |
| **Référence du test :** BoxChoice\_CreateBox\_V2 | | | **Nom du module :** createBoxes | | |
| **Objectifs du test :** Le but de ce test est de prouver que lorsqu’on clique sur une consigne la couleur change. | | | | | |
| **Conditions du test** | | | | | |
| **Logiciel :**  -Application test : createboxestest  -OS : Linux | | | | | |
| **Procédure de test** | | | | | |
| **Opérations** | Pour ce test, le programme se charge de créer 4 consignes et d’assigner à chacune un état (libre ou occupée) ainsi qu’une taille valide.  - étape 1 : Lancer le programme en tapant la commande  "./createboxestest test\_createBoxe\_UseNormal" dans un terminal.  - étape 2 : Vérifier que lorsqu’on clique sur une consigne celle-ci change de couleur. | | | | |
| **Résultats attendus** | | | | **Résultats obtenus** | |
| * changement de couleur d’une consigne : OK | | | | * OK | |
| **Conclusion :** OK | | | | | |