# Description des tests Fonctionnels

Table des matières

[Description des tests Fonctionnels 1](#_Toc442388873)

[I Introduction 2](#_Toc442388874)

[II Tests de l’insertion 2](#_Toc442388875)

[1) Tests de l’insertion (cas d’usage) 2](#_Toc442388876)

[2) Tests de l’insertion (cas critique) 3](#_Toc442388877)

[3) Tests de l’insertion (cas exeptionnel) 3](#_Toc442388878)

[II Tests de l’action HIT 4](#_Toc442388879)

[II Tests de l’action MOVE 4](#_Toc442388880)

[II Tests de l’action DELETE 5](#_Toc442388881)

[III Tests de l’action CLEAR 5](#_Toc442388882)

[IV Tests de l’action UNDO\REDO 6](#_Toc442388883)

[V Tests de l’action SAVE/LOAD 6](#_Toc442388884)

## I Introduction

Tous les tests effectués ont été remis dans le fichier test. Dans l’optique de mettre en place des tests de non regression nous avons modifié le script fourni pour obtenir des resultats aussi bien spécifique à chaque dossier que globaux.

## II Tests de l’insertion

### 1) Tests de l’insertion (cas d’usage)

Premierement le **DossierTest1** tend à tester trois choses:

* L’insertion des figures classique.
* L’insertion de figure composé.
* L’insertion recursive de composé qui contiennent des composés. Ces différents tests sont regroupés dans des dossiers Test:

|  |  |
| --- | --- |
| Test1 | Vérifie que le programme insére correctement un rectangle. |
| Test2 | Vérifie que le programme insére correctement un segment. |
| Test3 | Vérifie que le programme insére correctement un polygone. |
| Test4 | Vérifie que le programme construit et insére correctement un composite du type réunion. |
| Test5 | Vérifie que le programme construit et insére correctement un composite du type intersection. |
| Test6 | Vérifie que le programme construit et insére correctement un composite composé lui même de composite. |
| Test7 | Vérifie que le programme construit et insére correctement un composite prend seulement en compte les figures réelement présentes dans le modèle. |

### 2) Tests de l’insertion (cas critique)

Le **DossierTest2,** teste que le programme intéragit correctement avec l’utilisateur si celui ci n’entre pas les valeurs attendu par le programme.Ces différents tests sont regroupés dans des dossiers Test:

|  |  |
| --- | --- |
| Test 10 | Vérifie que le programme avertit l’utilisateur si le nom de la figure n’est pas renseigné. |
| Test 11 | Vérifie que le programme avertit l’utilisateur si les entrées désignant des points ne sont pas correctes.. |
| Test 12 | Vérifie que le programme n’insére pas le polygone si celui-ci n’est pas convexe. |
| Test 13 | Vérifie que le programme n’insére pas le polygone si celui-ci n’est pas strictement convexe. |
| Test 14 | Vérifie que le programme n’insére pas le polygone si le nombre de points renseigné ne correspond pas à la définition donné dans le cahier des charges. |
| Test 15 | Vérifie que le programme n’insére pas le rectangle si le nombre de points renseigné ne correspond pas à la définition donné dans le cahier des charges. |
| Test 16 | Vérifie que le programme n’insére pas le segment si le nombre de points renseigné ne correspond pas à la définition donné dans le cahier des charges. |
| Test 17 | Vérifie que le programme n’insére pas le composite si le nombre de figure est inférieur à deux figure. |

### 3) Tests de l’insertion (cas exeptionnel)

Le **DossierTest3** teste les cas d’insertion exeptionnel détaillé dans le cahier des charges.

|  |  |
| --- | --- |
| Test 16 | Vérifie que le nom est un identifiant unique de notre figure |
| Test 17 | Vérifie que les point du rectangle peuvent etre confondus |
| Test 18 | Vérifie que les point du segment peuvent etre confondus |
| Test 19 | Vérifie que les point du polygone peuvent etre confondus |

## II Tests de l’action HIT

Le **DossierTest4** teste la commande HIT.

|  |  |
| --- | --- |
| Test 20 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant sur l’arréte d’un polygone |
| Test 21 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant sur l’arréte d’un rectangle |
| Test 22 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant à l’intérieur d’un rectangle |
| Test 23 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant sur le segment |
| Test 24 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant à l’intérieur d’un polygone |
| Test 25 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant à l’intérieur d’une intersection dont le point appartenient à l’ensemble des figures |
| Test 26 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant à l’intérieur d’une réunion dont le point appartenient à l’ensemble des figures |
| Test 27 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant sur l’arréte d’un réunion dont le point n’appartenient pas à l’ensemble des figures |
| Test 28 | Vérifie que la méthode EstDedans() fonctionne pour un point se situant sur l’arréte d’une intersection dont le point n’appartenient pas à l’ensemble des figures |

## II Tests de l’action MOVE

Le **DossierTest5** teste la commande MOVE.

|  |  |
| --- | --- |
| Test 29 | Vérifie que la commande MOVE fonctionne pour un rectangle |
| Test 30 | Vérifie que la commande MOVE fonctionne pour un segment |
| Test 31 | Vérifie que la commande MOVE fonctionne pour un polygone |
| Test 32 | Vérifie que la commande MOVE fonctionne pour une réunion, de plus le test nous permet de vérifier que la copie qui s’effectue pour créer le composite est effectivement profonde |
| Test 33 | Vérifie que la commande MOVE fonctionne pour une intersection, de plus le test nous permet de vérifier que la copie qui s’effectue pour créer le composite est effectivement profonde |

## II Tests de l’action DELETE

Le **DossierTest6** teste la commande DELETE.

|  |  |
| --- | --- |
| Test 34 | Vérifie que la commande DELETE fonctionne pour une figure réelement présentes dans le modéle. |
| Test 35 | Vérifie que la commande DELETE uniquement la figure référencé par le nom placé en paramétre.. |
| Test 36 | Vérifie que la commande DELETE fonctionne pour un ensemble de figure présent dans le modèle. |
| Test 37 | Vérifie que la commande DELETE prévient l’utilisateur si la figure n’est pas présente dans le modéle. |
| Test 38 | Vérifie que la commande DELETE prévient l’utilisateur et ne supprime pas l’ensemble des figure si un nom référence une figure non présentes dans le modèle. |

## III Tests de l’action CLEAR

Le **DossierTest7** teste la commande CLEAR.

|  |  |
| --- | --- |
| Test 39 | Vérifie que la commande CLEAR efface toutes les figures du modéle. |
| Test 40 | Vérifie que la commande CLEAR avertit l’utilisateur si aucune figure n’est réelement présente dans le modéle. |

## 

## IV Tests de l’action UNDO\REDO

Le **DossierTest7** teste la commande UNDO\REDO.

|  |  |
| --- | --- |
| Test 41 | Vérifie que la commande UNDO avertit l’utilisateur si aucune action peut etre annulé. |
| Test 42 | Vérifie que la commande REDO avertit l’utilisateur si aucune action peut etre effectué, précédement annulé. |
| Test 43 | Vérifie que la création d’une figure peut etre undo ou redo. |
| Test 44 | Vérifie que la création la commande DELETE peut etre undo ou redo. |
| Test 45 | Vérifie que la création la commande CLEAR peut etre undo ou redo. (cf cahier des charges) |
| Test 46 | Vérifie que la série d’UNDO\REDO s’annule si une action modifie le modèle. |
| Test 47 | Vérifie que la création la commande MOVE peut etre undo ou redo. |

## V Tests de l’action SAVE/LOAD

|  |  |
| --- | --- |
| Test 48 | Vérifie que la commande SAVE fonctionne correctement. |
| Test 49 | Vérifie que la commande SAVEC fonctionne correctement. |
| Test 50 | Vérifie que la commande LOAD fonctionne correctement. |
| Test 51 | Vérifie que le programme avertis l’utilisateur si le sauvegarde du modéle est inutile car vide. |
| Test 52 | Vérifie que le programme avertis l’utilisateur si le chargement d’un fichier est inexistant dans le dossier courant. |