



Projet Annuel : 3i Architecture des logiciels

Kevin Maarek, Alexandre Fayette, Mathieu Péquin

# HoneyCode



Dossier de tests

## Table des matières

Tests unitaires .....	3
Test d'intégration.....	4
Introduction .....	4
Rapport .....	4
Tests d'intégration sous Windows .....	4
Tests d'intégration sous Debian Wheezy (7.3).....	5
Test d'acceptation .....	7
Introduction.....	7
Rapport.....	7

## Tests unitaires

La réalisation des tests unitaires se fait grâce à la librairie JUnit, qui nous permet de recenser les bugs les plus contraignants et apporter une dimension qualimétrique au projet. Ces tests nous permettent également de proprement structurer l'application et le code, en plus d'automatiser la phase de test d'intégration et de nous faire gagner un temps considérable malgré une augmentation de la charge de développement.

Les tests unitaires sont utilisés pour les classes de traitement (pas celles de définition d'interface graphique par exemple).

# Test d'intégration

## Introduction

Cette phase de test vise à valider le fonctionnement de l'application sous les différentes plateformes, suite au regroupement des modifications des développeurs.

Cette phase est dite manuelle, elle est donc à la charge d'un développeur, celui-ci devra valider les modifications de chacun de ses collègues. Il devra ensuite réunir l'ensemble de ces modifications et validé le bon fonctionnement de l'application.

Chacune des fonctionnalités de l'application devront être testées, qu'elles soient dépendantes ou non d'autres fonctionnalités.

Le testeur devra, après déroulement des tests, établir un rapport, signalant pour chacun des tests si ceux-ci on échoués ou non. Le testeur pourra y inclure des observations.

## Rapport

Les tests d'intégration ont été exécutés sous les principaux systèmes d'opérations, Microsoft Windows 7/8/8.1 et Linux Debian Wheezy (7.3) avec le Java Development Kit 7 update 55.

Par manque de moyens (financiers) ces tests n'ont pu être effectués sous l'environnement Apple Mac OS X.

Cependant un effort conséquent a été réalisé pour qu'HoneyCode soit pleinement fonctionnel sous Windows et Linux.

## Tests d'intégration sous Windows

F1 : Editeur de texte

-Ouverture/Enregistrement de fichiers source : Fonctionnel

-Edition de texte : Fonctionnel

-Création de workspaces (projets) : Fonctionnel

-Undo/Redo : Fonctionnel

- Copier/Couper/Coller : Fonctionnel

F2 : Outils pour l'édition de codes sources

-Colorisation syntaxique du code et thèmes : Fonctionnel

-Auto-complétion du code : Fonctionnel

F3 : Outils de développement

-Compilation du code Java: Fonctionnel

-Exécution du code, dans une console à part ou dans un encadré de sortie du programme :  
Fonctionnel (Manipulation à effectuer si console système)

F4 : Fonctionnement général

- Paramétrage de l'application : Fonctionnel
- Sauvegarde des paramètres : Fonctionnel
- Importation de plugins : Fonctionnel
- Arborescence des dossiers et fichiers du projet
- Fonctionnalités complémentaire à l'arborescence (créer dossier, renommer...) : Toutes fonctionnelles

Remarque :

Sous Windows, au premier lancement bien qu'un Java Development Kit soit bel et bien installé, il peut être nécessaire de spécifier le chemin du JDK, par exemple : C:\Program Files\Java\jdk1.7.0\_55.

## Tests d'intégration sous Debian Wheezy (7.3)

F1 : Editeur de texte

- Ouverture/Enregistrement de fichiers source : Fonctionnel
- Edition de texte : Fonctionnel
- Création de workspaces (projets) : Fonctionnel
- Undo/Redo : Fonctionnel
- Copier/Couper/Coller : Fonctionnel

F2 : Outils pour l'édition de codes sources

- Colorisation syntaxique du code et thèmes : Fonctionnel
- Auto-complétion du code : Fonctionnel

F3 : Outils de développement

- Compilation du code Java: Fonctionnel
- Exécution du code, dans une console à part ou dans un encadré de sortie du programme :  
Fonctionnel (Manipulation à effectuer si console système)

F4 : Fonctionnement général

- Paramétrage de l'application : Fonctionnel

- Sauvegarde des paramètres : Fonctionnel
- Importation de plugins : Fonctionnel
- Arborescence des dossiers et fichiers du projet
- Fonctionnalités complémentaire à l'arborescence (créer dossier, renommer...) : Toutes fonctionnelles sauf « Show in explorer » (ouvrir dans l'explorateur)

Sous certaines distributions Linux, l'API Desktop est très mal gérée ; ce qui entraîne la non fonctionnalité de « Show in explorer » (ouvrir dans l'explorateur). D'après nos recherches certaines librairies Gnome seraient à installer.

# Test d'acceptation

## Introduction

La phase de tests d'acceptation intervient après le développement, lorsque la phase de test d'intégration est terminée.

Les tests d'acceptation visent à vérifier que l'application répond correctement aux besoins et au cahier des charges.

Avec l'appui de l'équipe de développement, un utilisateur final vérifie dans l'ensemble, avec des scénarios réalistes que le rendu est conforme. Il peut donc questionner l'équipe de développement sur les différentes fonctionnalités de l'application.

Un feedback de l'utilisateur sera rendu et étudié par les développeurs. Ces derniers auront la possibilité de modifier l'application selon le compte-rendu de l'utilisateur.

## Rapport

Le livrable correspond et répond aux attentes des utilisateurs, aucun bogues gênants rencontrés.