



# CAHIER DE RECETTES

## Projet d'algorithmique 2016-2017

Version: CDR\_Groupe04\_RJA-OAK-ABE\_V1.4  
Auteur: Ouassim AKEBLI et Antoine BERENGUER

ISEN Toulon - Yncrea  
Maison du Numérique et de l'Innovation  
Place Georges Pompidou  
83000 Toulon

## Description du document

Type	Version	Confidentialité		
Cahier de recette	1.4	Usage externe		
Rédacteur	Nom	Fonction	Date	Visa
	Ouassim AKEBLI	Rédaction	05/01/17	
	Antoine BERENGUER		au 07/01/17	
	Vérificateur	Romain JACQUIEZ	Corrections	05/01/17
	Approbateur	Romain JACQUIEZ	Validation	12/01/17
Destinataire	Fonction	Organisme		
Client	Lecture	ISEN		



## Révisions du document

[illegible]

## Sommaire

1. INTRODUCTION.....	6
2. DESCRIPTION DES TESTS FONCTIONNELS.....	8
3. DESCRIPTION DES TESTS NON FONCTIONNELS.....	16
4. FICHE D'ANOMALIE.....	17

## Index des illustrations

## Index des tables

## REFERENCES

Référence	Description	Nom
[1]		
[2]		

## DEFINITIONS

Sans objet

## ABBREVIATIONS

ISEN : Institut Supérieur de l'Electronique et du Numérique

# 1. INTRODUCTION

La **recette d'un logiciel** consiste à vérifier si le produit obtenu à la fin du processus de développement est conforme aux attentes. Le cahier de recette décrit précisément les tests à réaliser pour réaliser la recette du logiciel.

Les concepts déjà abordés dans le cahier des charges se retrouvent liés aux concepts utilisés dans le cahier de recette. Ainsi nous pouvons dire:

- qu'une **exigence** dont il faut s'assurer du bon fonctionnement sera associée à
- un **cas de test** (ensemble des tests à réaliser afin de vérifier tout ou une partie d'une exigence), qui sera décrit dans
- une **fiche de tests** (description technique de la manière le cas sera vérifié, elle comporte le mode opératoire des opérations techniques à faire pour obtenir les résultats à vérifier), qui fera partie
- d'un **scénario** (ensemble de fiches de tests ordonnées avec une logique opératoire) associé à
- une **campagne de tests** (ensemble de scénarios exécutés sur un même environnement et par une même équipe) qui fera partie
- d'une **recette** (ensemble de campagnes de tests)

Il existe différents types de test , parmi lesquels :

- les **tests unitaires techniques**, qui sont réalisés par le programmeur, la plupart du temps à l'aide d'un cadriciel dédié comme CUnit ou JUnit, et qui permettent de valider un module fonction par fonction, afin de s'assurer qu'il est conforme aux spécifications fonctionnelles. Les test unitaires permettent notamment de vérifier que l'évolution du programme n'entraîne pas de régression.  
Ces tests s'appuient sur la connaissance précise des composants internes : boîte blanche.
- les **tests d'intégration**, qui permettent de valider le bon fonctionnement de toutes les parties du programme développées indépendamment.
- les **tests de performance**, qui portent sur le temps d'exécution et la consommation mémoire.
- les **tests de non régression** qui sont à effectuer lors de l'évolution du logiciel, afin de vérifier que ses fonctionnalités antérieures soient conservées à l'identique.
- les **tests de validation**, qui doivent vérifier que les exigences fonctionnelles et techniques ont été respectées.  
Ces tests ignorent la connaissance des composants internes : boîte noire.

La recette peut être:

- "**usine**", c'est à dire réalisée par le fournisseur du logiciel.  
Recette usine = tests unitaires + d'intégration + de validation + de non régression.

- ou bien "**utilisateur**", si elle est réalisée par le client.

La **recette fonctionnelle** validera les les spécifications exprimées dans le cahier des charges. La **recette technique** validera les caractéristiques techniques du produit livré, afin de s'assurer qu'il répond en terme de performances et d'exploitation à l'environnement dans lequel il sera utilisé.

Recette utilisateur = tests de validation + tests de performance.

## 2. DESCRIPTION DES TESTS FONCTIONNELS

L'identifiant de test sera de la forme TEST\_Ex\_xxxx\_x afin de permettre de se référer facilement à l'exigence concernée.

<b>Fiche de Test Fonctionnel : [TEST_Ef_0010_i] Démarrer une partie</b>				
<b>Objectif :</b>		Vérifier que le plateau s'affiche correctement		
<b>Exigence à tester :</b>		Ef_0010_i		
<b>Pré-requis :</b>		Programme fonctionnel		
<b>Initialisation :</b>		Démarrer le programme		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Lancer la partie		Affichage d'une grille de 5*5 cases	
<b>Rapport de test</b>		<b>Testé par :</b>		<b>Le :</b>
<b>Fonctionnalité</b>		<b>Conformité</b>		<b>Ergonomie</b>
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				



<b>Fiche de Test Fonctionnel : [TEST_Ef_0020_i] Vérification du score</b>				
<b>Objectif :</b>		Vérifier que le système de score fonctionne correctement		
<b>Exigence à tester :</b>		Ef_0020_i		
<b>Pré-requis :</b>		Programme complet et fonctionnel		
<b>Initialisation :</b>		Démarrer le programme		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Lancer la partie		Affiche un score du 0 à 0	
2	Faire une partie ou le joueur 1 gagne		Le score du joueur 1 s'incrémente de 1	
3	Faire une partie où le joueur 2 gagne		Le score du joueur 2 s'incrémente de 1	
<b>Rapport de test</b>		<b>Testé par :</b>	<b>Le :</b>	
<b>Fonctionnalité</b>		<b>Conformité</b>	<b>Ergonomie</b>	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

<b>Fiche de Test Fonctionnel : [TEST_Ef_0030_i] Recommencer une partie</b>				
<b>Objectif :</b>		Vérifier la possibilité de recommencer une partie		
<b>Exigence à tester :</b>		Ef_0030_i		
<b>Pré-requis :</b>		Programme complet et fonctionnel		
<b>Initialisation :</b>		Lancer le programme et démarrer une partie		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Faire mouvement		La grille est modifiée en fonction du mouvement	
2	Appuyer sur le bouton « recommencer la partie »		La grille se réinitialise	
<b>Rapport de test</b>		<b>Testé par :</b>		<b>Le :</b>
<b>Fonctionnalité</b>		<b>Conformité</b>		<b>Ergonomie</b>
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

Fiche de Test Fonctionnel : [TEST_Ef_0040_i] Quitter le programme				
<b>Objectif :</b>		Vérifier le bouton « quitter le programme » du menu principal		
<b>Exigence à tester :</b>		Ef_0040_i		
<b>Pré-requis :</b>		Programme avec menu principal fonctionnel		
<b>Initialisation :</b>		Lancer le programme		
Etapas du test :				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Cliquer sur le bouton « Quitter »		Sortie du programme	
<b>Rapport de test</b>		<b>Testé par :</b>	<b>Le :</b>	
<b>Fonctionnalité</b>		<b>Conformité</b>	<b>Ergonomie</b>	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

<b>Fiche de Test Fonctionnel : [TEST_Ef_0050_i] Réinitialisation du score</b>				
<b>Objectif :</b>		Test de la réinitialisation du score		
<b>Exigence à tester :</b>		Ef_0050_i		
<b>Pré-requis :</b>		Programme fonctionnel		
<b>Initialisation :</b>		Lancer le programme et démarrer une partie		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Faire une partie et gagner		Le gagnant remporte un point pour son score	
2	Cliquer sur le bouton « Réinitialiser le score »		Le score revient à 0-0	
<b>Rapport de test</b>		<b>Testé par :</b>		<b>Le :</b>
<b>Fonctionnalité</b>		<b>Conformité</b>		<b>Ergonomie</b>
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

<b>Fiche de Test Fonctionnel : [TEST_Ef_0080_i] Vérification des conditions de victoire</b>				
<b>Objectif :</b>		Possible de gagner par colonne, ligne ou diagonale		
<b>Exigence à tester :</b>		Ef_0080_i		
<b>Pré-requis :</b>		Programme complet et fonctionnel		
<b>Initialisation :</b>		Lancer le programme et démarrer une partie		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Jouer jusqu'à alignement vertical		Le joueur qui fait l'alignement remporte la partie	
2	Jouer jusqu'à alignement horizontal		Le joueur qui fait l'alignement remporte la partie	
...	Jouer jusqu'à alignement diagonal		Le joueur qui fait l'alignement remporte la partie	
<b>Rapport de test</b>		<b>Testé par :</b>	<b>Le :</b>	
<b>Fonctionnalité</b>		<b>Conformité</b>	<b>Ergonomie</b>	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

<b>Fiche de Test Fonctionnel : [TEST_Ef_0090_i] Vérification du changement de tour</b>				
<b>Objectif :</b>		Vérifie le système de changement de tour		
<b>Exigence à tester :</b>		Ef_0090_i		
<b>Pré-requis :</b>		Programme complet et fonctionnel		
<b>Initialisation :</b>		Lancer le programme et démarrer une partie		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Faire une tour en tant que joueur 1		Le jeu passe au joueur 2	
2	Faire une tour en tant que joueur 2		Le jeu passe au joueur 1	
<b>Rapport de test</b>		<b>Testé par :</b>	<b>Le :</b>	
<b>Fonctionnalité</b>		<b>Conformité</b>	<b>Ergonomie</b>	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

<b>Fiche de Test Fonctionnel : [TEST_Ef_0100_i] Vérification du placement de pion</b>				
<b>Objectif :</b>		Vérifier si il est possible de placer des pions correctement		
<b>Exigence à tester :</b>		Ef_0100_i		
<b>Pré-requis :</b>		Programme complet et fonctionnel		
<b>Initialisation :</b>		Lancer le programme et démarrer une partie		
<b>Etapes du test :</b>				
<b>Id</b>	<b>Démarche</b>	<b>Données</b>	<b>Comportement attendu</b>	<b>OK?</b>
1	Tenter de sélectionner un marqueur du joueur adverse		Aucune réaction du jeu	
2	Tenter de sélectionner un marqueur qui ne se trouve pas à l'extrémité du plateau		Aucune réaction du jeu	
3	Tenter de sélectionner un marqueur neutre		Le jeu propose de déplacer le marqueur et le marque de l'équipe appropriée après déplacement	
4	Idem que l'étape 3 mais avec un marqueur du joueur		Idem que étape 3	
<b>Rapport de test</b>		<b>Testé par :</b>	<b>Le :</b>	
<b>Fonctionnalité</b>		<b>Conformité</b>	<b>Ergonomie</b>	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

Fiche de Test Fonctionnel : [TEST_Ef_0110_i] Vérification de l'intelligence artificielle				
<b>Objectif :</b>		Vérifier si l'intelligence artificielle fonctionne		
<b>Exigence à tester :</b>		Ef_0110_i		
<b>Pré-requis :</b>		Programme complet et fonctionnel		
<b>Initialisation :</b>		Lancer le programme et démarrer une partie un joueur		
Etapas du test :				
Id	Démarche	Données	Comportement attendu	OK?
1	Faire un tour		Le jeu passe au joueur 2 et fait automatiquement son tour avant de repasser au joueur 1	
Rapport de test		Testé par :	Le :	
Fonctionnalité		Conformité	Ergonomie	
<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible		<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	<input type="checkbox"/> Excellente <input type="checkbox"/> Bonne <input type="checkbox"/> Moyenne <input type="checkbox"/> Faible	
<b>Commentaire :</b>				
<b>Fiches d'anomalies émises :</b>				
<b>Approbation :</b>				

### 3. DESCRIPTION DES TESTS NON FONCTIONNELS

Identifiant de test sera de la forme TEST\_Ex\_xxxx\_x afin de permettre de se référer facilement à l'exigence concernée.

Identifiant	Exigences	Description	OK ?
TEST_Ed_0010_i	Ed_0010_i	Avons-nous une interface graphique cliquable ?	
TEST_Ed_0020_i	Ed_0020_i	Le plateau est-il modélisé par une grille ?	
TEST_Ed_0030_i	Ed_0030_i	Avons-nous un bouton jouer fonctionnel dans le menu principal ?	



TEST_Ed_0040_i	Ed_0040_i	Avons-nous un bouton crédits fonctionnel dans le menu principal ?	
TEST_Ed_0050_i	Ed_0050_i	Avons-nous un bouton règles fonctionnel dans le menu principal ?	
TEST_Ed_0060_i	Ed_0060_i	Avons nous un bouton pour revenir au menu principal lors d'une partie ?	
TEST_Ed_0070_i	Ed_0070_i	Les deux joueurs ont-ils respectivement le symbole X et O pour les représenter ?	
TEST_Ed_0080_i	Ed_0080_i	Peut-on redimensionner la fenêtre sans avoir de problèmes graphique ?	
TEST_Ed_0090_i	Ed_0090_i	Les couleurs principales du jeu sont-elles le blanc, marron et marron foncé ?	
TEST_Ed_0100_i	Ed_0100_i	Est-ce que sélectionner un cube montre en évidence les déplacements possibles ?	
TEST_Ed_0110_i	Ed_0110_i	Clic gauche permet-il de sélectionner et valider ?	
TEST_Ed_0120_i	Ed_0120_i	Clic droit permet-il d'annuler une sélection ?	
TEST_Ed_0130_i	Ed_0130_i	Les cubes à l'extrémité sont-ils les seuls affectés par le système de sélection ?	
TEST_Ei_0010_i	Ei_0010_i	Le programme fonctionne-t-il sous GNU/Linux ?	
TEST_Ei_0020_i	Ei_0020_i	Le programme respecte-t-il la norme C99 ?	
TEST_Ep_0010_i	Ep_0010_i	Le programme est-il léger au niveau consommation mémoire ?	
TEST_Ep_0020_i	Ep_0020_i	Le score des joueurs est-il limité à 99 ?	
TEST_Em_0020_i	Em_0020_i	Les sources du code sont-elles commentées ?	
TEST_El_0010_i	El_0010_i	Le programme s'appelle-t-il Quixisen ?	

## 4. FICHE D'ANOMALIE

Identifiant d'anomalie sera de la forme ANO\_TEST\_Ex\_xxxx\_x afin de permettre de se référer facilement au test concerné.

<b>Fiche d'anomalie : [Identifiant de l'anomalie]</b>	
<b>Type :</b>	Description du type d'anomalie
<b>Gravité :</b>	Indication de la gravité de l'anomalie
<b>Description de l'anomalie</b>	
<b>Fiche de test :</b>	
<b>Comportement attendu</b>	
<b>Comportement constaté</b>	

<b>Analyse du problème :</b>