



# CAHIER DES CHARGES

## Projet d'algorithmique 2016-2017

Version: CDC\_Groupe04\_GDE\_MBO\_PPA\_V1.3

Auteur: ISEN Toulon

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

## Description du document

Type		Version	Confidentialité	
Cahier des charges		1.3	Usage externe	
	Nom	Fonction	Date	Visa
	Rédacteur	ISEN Toulon		
	Vérificateur			
	Approbateur			
Destinataire		Fonction	Organisme	
Public			ISEN	



## Révisions du document

Version	Date	Rédacteur	Modifications
1.0	18/07/2016	FMC	Mise en forme
1.1	06/01/2017	PPA / MBO / GDE	Mise en place des exigences
1.2	11/01/2017	MBO / GDE	Ajout de l'IA
1.3	14/01/2017	GDE	Finalisation exigences

## Sommaire

1. Présentation du projet.....	7
A) Contexte du projet.....	7
B) Objectifs du projet.....	8
C) Description de l'existant.....	8
D) Acceptation du produit.....	8
2. Expression fonctionnelle des besoins.....	8
A) Besoins fonctionnels.....	9
B) Besoins non fonctionnels.....	12
3. Contraintes.....	13
A) Budgétaires.....	13
B) Temporelles.....	13
C) Normatives.....	13
4. Déroulement du projet.....	13
A) Planification.....	13
B) Documentation.....	13
C) Équipe projet et responsabilités.....	13

Ce document est un exemple de cahier des charges pour un projet de développement logiciel. Le cahier des charges formalise le besoin afin qu'il soit compris par l'ensemble des personnes qui vont être impliquées dans le projet. Il doit indiquer précisément les spécifications attendues.

Pour qu'un projet puisse être qualifié de réussi, il est impératif de définir précisément :

- les objectifs fixés,
- les ressources requises,
- la planification,
- les méthodes d'évaluation,
- les méthodes de contrôle.

Considéré dans sa globalité, un projet requiert les étapes suivantes :

- nomination d'une équipe et d'un responsable du projet,
- état de l'art et exploration des possibilités techniques,
- rédaction du cahier de charges
- mise en œuvre, suivi et évaluation du projet.

La structure suivie dans ce document s'inspire de ce que propose la norme NF EN 16271 (février 2013) qui a remplacé la norme NF X50-151 (septembre 2007).

*"Le cahier des charges est un support indispensable de dialogue entre l'utilisateur et le concepteur. L'intérêt essentiel pour les deux parties est de ne pas découvrir des non-conformités fonctionnelles, d'usage ou réglementaires après réalisation. Il est en effet plus facile et moins coûteux d'agir dès la conception. "* INRS, ED6231, mai 2016.

## Index des illustrations

Illustration 1: Plateau de jeu Quixo.....	7
Illustration 2: Fonctionnement du Quixo.....	7

## Index des tables

## REFERENCES

Référence	Description	Nom
[1]		
[2]		

## DEFINITIONS

Sans objet

## ABBREVIATIONS

ISEN	:	Institut Supérieur de l'Electronique et du Numérique
SQL	:	Structured Query Language

## 1. Présentation du projet

### A) Contexte du projet



Illustration 1: Plateau de jeu Quixo

Le jeu du Quixo a connu sa première version commerciale en 1995. C'est un jeu de réflexion sur un plateau de 5\*5 cubes. Deux adversaires sont opposés et ont chacun pour objectif d'aligner (ligne ou diagonale) 5 cubes de même symbole.

Les 25 cubes sont prisonniers du plateau : chaque cube possède 4 faces neutres, une face marquée d'une croix et une face marquée d'un rond. Au départ, les cubes sont disposés faces supérieures neutres. Chacun son tour, le joueur choisit un cube situé sur la partie extérieure du plateau. Il le replace avec sa marque sur la face supérieure, en poussant sur une rangée incomplète (principe du pousse – pousse).

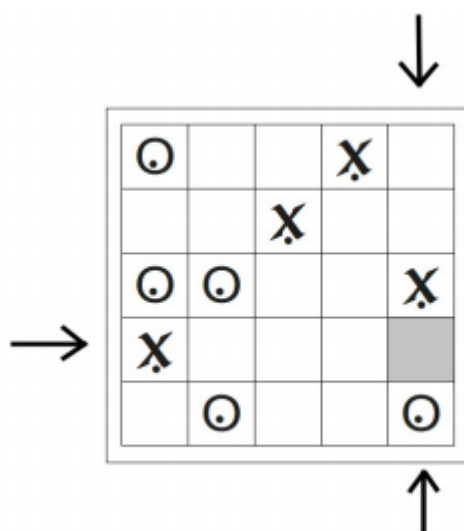


Illustration 2: Fonctionnement du Quixo

Un mode 2 contre 2 existe aussi, alliant alors les faces marquées de croix aux faces marquées d'un rond.

Le jeu est conçu avec des cubes en bois engendrant une consommation abusive de bois ce qui peut nuire à l'environnement.

## **B) Objectifs du projet**

L'objectif du projet est de proposer une version logicielle du Quixo, pouvant se jouer seul contre l'ordinateur, à deux sur le même ordinateur ou à quatre, en deux contre deux sur le même ordinateur. Cela permettra de réduire à zéro le nombre de cubes en bois à produire et permettra également d'organiser des tournois entre joueurs sans avoir besoin du matériel physique adéquat.

## **C) Description de l'existant**

Il n'y a pas de concurrents sur ce secteur, notre produit sera donc totalement innovant.

## **D) Acceptation du produit**

Le logiciel sera considéré comme acceptable s'il vérifie tous les tests spécifiés dans le cahier de recette.

# **2. Expression fonctionnelle des besoins**

Les exigences décrites ci-après ont pour but de décrire les caractéristiques du projet.

Elles sont verbalisées en trois parties : objet-verbe-attribut, où :

- objet désigne le sujet sur lequel porte l'exigence ;
- attribut désigne la caractéristique que le verbe impose à l'objet.

## **Numérotation**

Les exigences sont numérotées sur quatre chiffres, de dix en dix pour pouvoir éventuellement insérer, après accord de toutes les parties concernées, de nouvelles exigences à côté de précédentes exigences.

Un numéro d'exigence abandonné ne pourra pas être réutilisé.

Les exigences sont de plusieurs types, chaque type décrit ci-après étant représenté par une lettre caractéristique. Cette lettre prefixera chaque exigence.

Les exigences peuvent être facultatives ou indispensables. Les lettres F et I suffixeront respectivement les numéros d'exigences décrites ci-dessus.

## **Types d'exigences**

Les types d'exigences sont les suivants (avec entre parenthèses la lettre servant de préfixe au numéro d'exigence) :



- **exigences Fonctionnelles (EF)** : portent sur ce que le produit doit être capable de faire ;
- **exigences de Design (ED)** : portent sur tout ce qui a trait à la forme au travers de laquelle on interagira avec le produit ;
- **exigences d'Implémentation (EI)** : portent sur les contraintes techniques liées à la réalisation du produit ;
- **exigences de Performance (EP)** : portent sur des niveaux quantitatifs qui doivent être atteints dans des conditions à préciser ;
- **exigences de Maintenance (EM)** : portent sur tout le support qui est à mettre en œuvre pour assurer le bon fonctionnement du produit ;
- **exigences de Validation (EV)** : portent sur les actions qui permettent de valider des exigences ;
- **exigences Sécuritaires (ES)** : portent sur les mesures à prendre pour assurer la sécurité et la sûreté des informations ;
- **exigences Légales (EL)** : portent sur les mesures à prendre pour assurer la légalité des actions du produit.

Exemple : EF\_0010\_I sera l'exigence fonctionnelle indispensable 0010.  
EP\_0030\_F sera l'exigence de performance facultative 0030.

## A) *Besoins fonctionnels*

Les besoins fonctionnels décrivent les différentes fonctions que le logiciel devra assurer. Il est préférable de les classer par catégories, afin d'en simplifier la lecture.

### Exigences fonctionnelles

EF\_0010\_I

Les joueurs joueront sur une grille de 5 cases par 5 cases.

EF\_0020\_I

Chaque joueur choisit un symbole (croix ou cercle).

EF\_0030\_I

Chaque cube possédera trois faces différentes (croix, rond et neutre).

EF\_0040\_I

Chaque joueur à tour de rôle choisit un cube neutre ou de son signe qui se trouve à l'extrémité de la grille.

EF\_0050\_I

Chaque joueur repose son cube à l'extrémité d'une rangée incomplète.

EF\_0060\_I

Un joueur ne peut pas piocher un cube situé sur les 9 cases centrales.

EF\_0070\_I

Le joueur ne peut reposer un cube à l'endroit où il l'a pioché.

EF\_0080\_I

Le joueur ne peut pas piocher un cube ayant déjà été joué par l'adversaire.

EF\_0090\_I

Un joueur gagne lorsqu'il a aligné cinq mêmes faces de son symbole.

EF\_0100\_I

Si un joueur crée involontairement une ligne gagnante de signe adverse, alors il perd, même si dans le même temps il a réussi à compléter la sienne.

EF\_0110\_I

Un joueur ne peut pas passer son tour.

EF\_0120\_I

Pour activer un bouton, l'utilisateur utilisera un clic gauche de souris.

EF\_0130\_I

L'utilisateur accédera au début du jeu à un menu principal.

EF\_0140\_I

Le joueur pourra revenir au menu principal lors d'une partie.

EF\_0150\_I

Le joueur pourra quitter le jeu à partir de l'écran principal.

EF\_0160\_I

Le joueur, à partir de l'écran principal pourra accéder aux règles du jeu.

EF\_0161\_I

Le joueur, à partir du menu des règles pourra revenir au menu principal.

EF\_0170\_I

Le joueur, à partir de l'écran principal pourra accéder au jeu en mode une personne contre une autre classique.

EF\_0180\_I

Le joueur, à partir de l'écran principal pourra accéder au jeu en mode une personne contre l'ordinateur.

EF\_0190\_I

En fin de partie le joueur accède à un menu affichant le gagnant et les scores.

EF\_0191\_I

Lors de l'affichage du gagnant et des scores, le joueur pourra recommencer une partie en gardant les scores des joueurs.

EF\_0192\_I

Lors de l'affichage du gagnant et des scores, le joueur pourra revenir au menu principal.

EF\_0200\_F

A partir de l'écran principal, le joueur pourra accéder au jeu en mode 2 personnes contre 2 autres.

EF\_0210\_F

Un mode en équipe de 2 joueurs contre 2 joueurs sera disponible.

EF\_0220\_I

Un joueur pourra jouer en mode 1 personne contre l'ordinateur sur le mode classique du Quixo.

EF\_0230\_I

L'ordinateur sera capable de déterminer des coups à jouer lors du mode 1vs1A.

## Exigences de design

ED\_0010\_I

Le jeu s'affichera dans une fenêtre de taille 800x800.

ED\_0020\_I

Le jeu permettra de sélectionner un cube et le déplacer par le clic gauche.

ED\_0021\_I

Le jeu affichera les zones possibles de pose du cube en surbrillance.

ED\_0030\_I

Le jeu affichera la face neutre des cubes en début de partie.

ED\_0040\_I

Les trois symboles sur les faces sont croix ou rond ou neutre.

ED\_0050\_I

Le jeu affichera une case de couleur différente pour représenter une case vide.

ED\_0060\_I

L'utilisateur pourra modifier la taille de la fenêtre. Les éléments resteront en place.

ED\_0070\_I

L'utilisateur pourra suivre son score au fil des parties.

ED\_0080\_I

En fin de partie le joueur accède à un menu affichant le gagnant et les scores.

ED\_0081\_I

Un bouton « Rejouer » lors de l'affichage du gagnant et des scores permettra de recommencer une partie en gardant les scores des joueurs.

ED\_0082\_I

Un bouton « Menu » lors de l'affichage du gagnant et des scores permettra de revenir au menu principal.

ED\_0090\_I

Un bouton « Menu » permettra de revenir au menu lors d'une partie.

ED\_0100\_I

Affichage du joueur courant lors d'une partie.

ED\_0110\_I

Au début d'une partie, les joueurs choisiront leurs symboles.

ED\_0120\_I

Le fond sera de couleur grise (#EFF0F4).

ED\_0121\_I

Les boutons seront de couleur bleu (#415F9D).

ED\_0122\_I

Les cubes seront de couleur bleu (#233B6E).

ED\_0123\_I

Les symboles seront de couleur grise (#EFF0F4).

ED\_0124\_I

Le texte sera de couleur grise (#EFF0F4) sur bleu et bleu (#415F9D) sur gris.

ED\_0130\_I

Le jeu sera entièrement joué à la souris à l'aide des clics gauches.

ED\_0140\_I

Un bouton « Quitter » sur l'écran principal permettra de quitter le jeu.

ED\_0150\_I

Un bouton « Règles » sur l'écran principal permettra d'accéder aux règles du jeu.

ED\_0151\_I

Un bouton « Menu » sur l'écran des règles permettra de revenir au menu principal.

ED\_0160\_I

Un bouton « 1 vs 1 » sur l'écran principal permettra d'accéder au jeu en mode 1 personne contre une autre classique.

ED\_0170\_I

Un bouton « 1 vs IA » sur l'écran principal permettra d'accéder au jeu en mode 1 personne contre l'ordinateur.

ED\_0180\_I

L'utilisateur accédera au début du jeu à un menu principal.

ED\_0190\_F

Un bouton « 2 vs 2 » sur l'écran principal permettra d'accéder au jeu en mode 2

personnes contre 2 autres.

## **B) Besoins non fonctionnels**

Les besoins non fonctionnels sont tous ceux qui n'ont pas trait aux fonctions du logiciel, comme le système d'exploitation sur lequel il devra fonctionner.

### **Exigences d'implémentation**

EI\_0010\_I

Le logiciel devra fonctionner sous système d'exploitation Linux.

EI\_0020\_I

Le logiciel devra être écrit en langage C99.

### **Exigences de performance**

EP\_0010\_I

Le logiciel devra consommer au plus 200 Mo de mémoire.

EP\_0020\_I

La phase de réflexion de l'ordinateur ne devra pas excéder 20 secondes.

### **Exigences de maintenance**

EM\_0010\_I

Le logiciel sera maintenu par l'équipe projet pendant 4 mois.

EM\_0020\_I

Les sources du logiciel seront commentées au format Doxygen.

### **Exigences de validation**

EV\_0010\_I

L'interface utilisateur sera validée manuellement.

### **Exigences sécuritaires**

Aucune exigence sécuritaire.

### **Exigences légales**

Aucune exigence légale.

### 3. Contraintes

#### **A) Budgétaires**

Le budget alloué pour ce projet sera de 30 000 €. Les moyens matériels mis à disposition sont trois ordinateurs reliés à internet.

#### **B) Temporelles**

La livraison du produit fini est prévue en mars 2017. La documentation relative à l'analyse du logiciel devra être livrée fin janvier 2017.

#### **C) Normatives**

Le code source du projet devra respecter la norme C99.

### 4. Déroulement du projet

#### **A) Planification**

Le projet se déroulera sur 4 mois.

La première semaine sera consacrée à l'analyse de l'existant et à la rédaction du CDC, du CDR, du CDCG et du CDCD.

#### **B) Documentation**

Le projet livré sera accompagné d'un cahier de conception général et détaillé, d'un cahier de recette, et d'un rapport final.

#### **C) Équipe projet et responsabilités**

L'équipe sera constituée d'un chef de projet, et de 2 développeurs.