

# Stratégie de tests T3

Groupe JAL22-T3-C / Great Wise Médecins

NOEL Arnaud - NAEGELY Jeannot - JOLY Dorian - DEBEZE Isaïe

## Introduction :

Ce livrable a été réalisé sur demande d'un rendu de dossier de stratégie de tests dans le contexte du projet tuteuré T3. Son objectif est de nous faire réaliser l'importance de la phase de test d'une application, et de nous laisser en faire l'expérience au travers de celui-ci.

Faisant partie du groupe de TP numéro 3, le sujet que nous devons réussir à faire comprendre au joueur de notre jeu est "L'importance de la prise en compte des antécédents et du contexte de vie des patients lors du diagnostic médical".

Aucun attendu n'a été précisé au préalable, que ce soit au sujet des performances ou de la sécurisation du logiciel. Les fonctionnalités devraient permettre au joueur de comprendre l'objectif pédagogique et les interfaces devraient être instinctives dans leur navigation. Nous pouvons donc décider de quels résultats seront à atteindre, notamment pour avoir des références lors des tests.

## **1ère partie :**

La personne responsable de nous donner les modalités du projet tuteuré, Mr. GOSSA, ne nous a pas donné d'instructions ou de prérequis au niveau des exigences de tests sur le projet, il nous est donc par conséquent difficile de remplir cette partie.

Cependant, nous pouvons établir nos propres exigences vis-à-vis de notre production. Si nous donnons des objectifs pour chaque type de test, nous pouvons synthétiser ça de la manière suivante.

### **Tests fonctionnels :**

Toutes les fonctionnalités doivent être utilisables, et si elles ne le sont pas parfaitement, elles ne doivent pas affecter le déroulement interne de la partie. Nous accepterons au maximum des problèmes d'affichages légers.

### **Tests IHM :**

Le joueur doit pouvoir rapidement comprendre l'interface pour pouvoir facilement utiliser le logiciel et assimiler le déroulement d'une partie. Nous demanderons donc à des camarades de tester le jeu pour savoir ce qui est compréhensible ou non.

### **Tests de performances et de charge :**

Notre application n'est pas vraiment concernée par les tests de charges étant donné que l'utilisateur est seul sur son instance du jeu et qu'il ne peut pas y insérer de données nouvelles. Nous ferons quelques tests de performances pour voir quels temps d'exécution pourraient être améliorés.

### **Tests de sécurité :**

Notre jeu ne gère pas la sauvegarde (et la reprise) et est inaccessible en tant que tel car le jeu se joue en "solo" (joueur unique) sur le jeu téléchargé. Nous ne sommes donc pas vraiment concernés par cette catégorie (plus de précisions en 2ème partie).

## **2ème partie :**

### **Tests fonctionnels :**

- Fonctions métiers : la vision « Utilisateurs »

Il faut que chaque fonctionnalité soit évaluée comme fonctionnelle et ne posant pas de problème dans le déroulement de la partie.

Dans notre cas, nous allons faire tester à plusieurs personnes extérieurs à notre projet pour avoir une vision neutre ou “Utilisateurs”.

Temps: 30 minutes

- Prise en compte des accès concurrents

C’est impossible dans notre projet car le jeu est dit “solo” et donc il ne peut pas y avoir plusieurs personnes en même temps sur notre jeu.

Temps: X

- Limites de fonction

Dans notre cas, cette limite sera impossible car nous n'avons aucun élément que le joueur peut insérer depuis l'application.

Temps: X

- Intégration avec d’autres logiciels

Notre application n'utilise qu'un seul logiciel et ne doit intégrer aucun autre logiciel.

Temps: X

- Fonctionnement multi plate-formes

Nous avons uniquement un exécutable et un installateur pour Windows.

Mais dans le cas où nous ferions différents exécutables, nous pourrions tester notre application sur plusieurs plateformes comme Windows, Linux, MacOS...

Temps: 20 minutes

## **Tests IHM :**

- Facilité d'utilisation

Ces tests d'utilisabilité sont très importants. L'apprentissage du logiciel doit être simple et court. Pour notre part, on va voir si le principe du jeu est vite compris et si l'interface n'est pas trop compliquée à comprendre.

Temps: 20 minutes

- Cohérence / Cohésion (charte graphique)

Pour cette partie, nous allons voir si les couleurs de toutes les fenêtres sont cohérentes, si tous les textes ont la même police, les fonds sont en accord.

Temps: 10 minutes.

- Messages et comportements sur erreurs

Il y en a un dans notre application donc on va le tester pour voir s'il maintient la bonne fonction qu'on lui a attribuée.

Temps: 5 minutes

- Chargements initiaux

Nous avons 4 écrans de jeu différents. Nous allons donc tester ces 4 écrans pour voir s'ils mettent du temps à apparaître.

Temps: 10 minutes

## **Tests de performances et de charges :**

- Analyse des performances - Temps de réponses

Dans notre application, nous pouvons analyser les performances de notre code, nos fonctions, nos algorithmes avec, pourquoi pas, des timer dans notre code pour voir en combien de temps s'exécute telles et telles fonctions.

Temps: 30 minutes

- Tests de volumes

Dans notre cas, le test de volume ne sera pas important car nous utilisons des données fixes et non variables dans l'application.

Temps: X

- Tests de surcharge

Dans notre cas, aucune donnée est variable (en tout cas pas en nombre) donc une surcharge est impossible.

Temps: X

- Charges serveur, bande passante, débit réseau, ...

Dans notre cas, aucun serveur n'est utilisé car notre application n'est pas en ligne ou multijoueur donc aucune connexion n'est requise (à part pour télécharger l'application).

Temps: X

## **Tests de sécurité :**

- Habitations

Nous ne mettons pas à disposition du joueur de choix de rôle dans l'application, l'expérience de jeu n'étant pas censée changer du point de vue du médecin.

Temps: X

- Accès externes

L'application n'étant pas connectée en réseau (le jeu n'est pas multijoueur), aucun accès externe à l'application n'est possible pendant une partie.

Temps: X

- Fonctionnement en mode dégradé

Notre application n'intègre que des fonctionnalités primordiales à son bon fonctionnement, avoir un mode dégradé de celle-ci irait donc à l'encontre de la compréhension de l'objectif pédagogique.

Temps: X

- Sauvegarde et reprise

Aucun moyen de sauvegarde de jeu n'est implémenté, car le temps d'une partie ne le nécessite pas (cela permet également la concentration du joueur, puisqu'il ne peut pas sauvegarder et revenir en arrière en cas d'erreur).

Temps: X

•PCA/PRA

La sauvegarde n'étant pas possible, il nous est impossible et inutile de tester la continuation et la reprise d'une partie de jeu.

Temps: X

**Matrice fonctionnelle :**

		Importance pour l'utilisateur		
		Stratégique	Importante	Secondaire
<b>F r é q u e n c e</b>	<b>Forte</b>	<ul style="list-style-type: none"><li>- Ouvrir l'encyclopédie</li><li>- Tourner les pages de l'encyclopédie</li><li>- Questionner le patient</li></ul>		
	<b>Moyenne</b>	<ul style="list-style-type: none"><li>- Diagnostiquer manuellement</li><li>- Quitter la partie</li></ul>	<ul style="list-style-type: none"><li>- Afficher le tutoriel</li><li>- Relancer une partie depuis l'écran de fin</li></ul>	<ul style="list-style-type: none"><li>- Afficher le résultat du diagnostic</li></ul>
	<b>Faible</b>	<ul style="list-style-type: none"><li>- Lancer une partie</li></ul>	<ul style="list-style-type: none"><li>- Quitter le jeu</li></ul>	<ul style="list-style-type: none"><li>- Afficher les crédits</li><li>- Mise en plein écran (paramètres)</li></ul>

**Regroupement des fonctionnalités ordonnées par importance:**

- **Tests de catégorie 1:** Ouvrir l'encyclopédie, Tourner les pages de l'encyclopédie, Questionner le patient, Diagnostiquer manuellement, Quitter la partie
- **Tests de catégorie 2:** Lancer une partie
- **Tests de catégorie 3:** Afficher le tutoriel, Relancer une partie depuis l'écran de fin
- **Tests de catégorie 4:** Quitter le jeu, Afficher le résultat du diagnostic, Afficher les crédits, Mise en plein écran (paramètres)

## 3ème partie :

### Moyens de test :

Le jeu est testé sur des ordinateurs fonctionnant sous Windows et ayant des performances graphiques et moteurs de très peu à très élevées.

Seul un nombre réduit de machines est nécessaire, entre 6 et 8, car le programme est très peu gourmand en ressources.

Pour tester le jeu nous pouvons essayer de le pousser vers ses limites :

Pour les différentes typologies de test, certaines méthodes peuvent être adoptées :

- IHM, nous pouvons demander à des personnes extérieures de juger les interfaces et la cohésion du jeu
- Fonctionnels, nous pouvons vérifier les fonctionnalités une par une puis dans leur ensemble
- Performance / Charge :
  - Jouer des parties en essayant d'atteindre des valeurs maximales
  - Jouer des parties en suivant des parcours d'actions inhabituels
  - Tester des valeurs limites en les insérant directement depuis le code source
- Sécurité, pas vraiment de test pour notre application

### Estimation de la durée de la campagne de tests :

Type de test	Temps de test (minutes)
Fonctionnels	50
IHM	45
Charge/Perf.	30
Sécurité	0
Total	125

Nous estimons donc que dans un bon déroulement nos tests prendraient environ 2h 05min à être complétés.



**Outillage des tests :**

Le modèle de fiche de test de notre projet se trouve dans le dossier de rendu, sous le nom "*Fiche\_Test\_DEBEZE\_JOLY\_NAEGELY\_NOEL.pdf*".

## Conclusion :

Ce dossier nous a permis d'identifier les différents tests à réaliser sur notre application ainsi que le temps qu'il faudrait utiliser pour les mettre en œuvre. Nous avons pu ainsi constater l'importance de réaliser de tels tests lors du développement d'un projet de cette ampleur, mais également imaginer le volume de tests sur des applications plus complètes.

Malgré leur coût conséquent en temps, nous avons pu déduire que sans tests, il est possible de se retrouver avec une application instable ne répondant pas aux exigences de base.

Une chose un peu dommage est que nous avons écrit ce dossier de test après avoir testé et fait tester notre jeu, ce qui permet uniquement de s'imaginer et non de faire l'expérience du déroulement des tests plus formalisés présents dans ce dossier.

### Tableau récapitulatif de consommation de temps :

Prénom	Temps en TD/TP (heures)	Temps à la maison (heures)	Temps total (heures)
Jeannot	6	-	6
Dorian	6	-	6
Isaïe	6	2	8
Arnaud	6	-	6
Total	24	2	26