



Plan de tests logiciels

Version 2.0

Historique des révisions

Date	Version	Description	Auteur
2020-03-20	1.0	Première rédaction du plan de tests	Duc-Thien Nguyen
2020-04-13	2.0	finalisation du plan de tests	François-Xavier Legault

Table des matières

1. Introduction	4
2. Exigences à tester	4
3. Stratégie de test	5
3.1. Types de test	5
3.1.1. Tests de fonction	5
3.1.2. Tests d'interface usager	5
3.1.3. Tests d'intégrité des données	5
3.1.4. Tests de performance	6
3.1.5. Tests de charge	6
3.1.6. Tests de stress	6
3.1.7. Tests de volume	7
3.1.8. Tests d'échec/récupération	7
3.2. Outils	8
4. Ressources	8
4.1. Équipe de test	8
4.2. Système	9
5. Jalons du projet	9

Plan de tests logiciels

1. Introduction

Ce document est utilisé pour mettre en place des manières pour s'assurer que les logiciels conçus durant la session sont conformes aux exigences demandés. Ces exigences sont initialement données dans le complément pédagogique et ils sont développés en profondeur dans le document SRS. En planifiant les stratégies de tests, nous nous assurons que les applications ne vont pas avoir des comportements indésirables, par exemple que le logiciel arrête de fonctionner soudainement. Nous allons tout d'abord commencer par une énumération des exigences à tester, ensuite démontrer certaines stratégie de test, aussi présenter nos ressources pour effectuer les tests et enfin, une description des jalons.

2. Exigences à tester

Exigences	Tests associés
<ul style="list-style-type: none">• Clavardage - Intégration	Test d'intégrité des données, test de fonction
<ul style="list-style-type: none">• Clavardage - Canaux de discussion	Test de charge, test de fonction
<ul style="list-style-type: none">• Profil d'utilisateur et historique	Test de fonction, test d'intégrité des données, test d'interface usager
<ul style="list-style-type: none">• Sélection et importations des avatars	Test de fonction
<ul style="list-style-type: none">• Photos de profils	test de charge
<ul style="list-style-type: none">• Personnalités des joueurs virtuels	test de fonction
<ul style="list-style-type: none">• Création d'un jeu	Test de stress, Test de charge, test de fonction, test d'interface usager
<ul style="list-style-type: none">• Les types de parties privés	test de volume, test d'interface usager
<ul style="list-style-type: none">• Modes de jeu	Test d'interface usager, test de fonction
<ul style="list-style-type: none">• Utilisation de l'application générale	Test d'interface usager, test de performance
<ul style="list-style-type: none">• Performance	Test de stress, test de charge

3. Stratégie de test

3.1. Types de test

3.1.1. Tests de fonction

Objectif de test:	Vérifier le bon fonctionnement des fonctions des logiciels selon les exigences détaillés dans le document SRS.
Technique:	Utiliser les logiciels en suivant le “flow” de jeu et observer les comportements s’ils sont acceptables selon le SRS.
Critère de complétion:	Les résultats suivant les actions faites par un utilisateur doit être conforme avec le SRS.
Considérations spéciales:	

3.1.2. Tests d’interface usager

Objectif de test:	Vérifier si l’interface usager est facilement navigable selon la série d’action d’un usager.
Technique:	Pour le clients lourds et léger, le testeur doit essayer de détruire les logiciels en faisant n’importe quoi sur l’interface (ex: cliquer partout, cliquer sur des boutons à répétitions, entrées des données dans des endroits spécifiques, etc.). Pendant les actions sur les interfaces, le testeur doit observer s’il y a des séries d’actions qui présente une mauvaise expérience d’utilisateur.
Critère de complétion:	Si toutes les navigations et les entrées de données ne font pas arrêter le logiciel soudainement ainsi que l’affichage de l’interface conforme selon la navigation.
Considérations spéciales:	

3.1.3. Tests d’intégrité des données

Objectif de test:	Vérifier que les entrées de données et les effacements.
Technique:	Faire des séries de requêtes vers le serveur sur toutes les routes reliant avec la base données. Pour certaines requêtes, elles doivent avoir des informations valides et invalides afin de vérifier que les invalides ne sont pas utilisés.
Critère de complétion:	La base de données contient les informations avec les requêtes contenant des informations valides seulement. Aussi, les informations effacées ne doivent pas être présentes.
Considérations spéciales:	La communication entre le testeur et la base de données doit être passée par le serveur.

3.1.4. Tests de performance

Objectif de test:	Vérifier que les applications peuvent fonctionner correctement même s'il y a beaucoup de trafic dans le serveur.
Technique:	Plusieurs testeurs doivent être connectés simultanément. Les joueurs vont essayer de jouer sur les applications comme normalement. Pendant le jeu, des testeurs vont aussi "spam" les canaux de discussions. Les testeurs vérifient s'il y a des long moment d'attentes ou des temps d'attente insatisfaisant qui pourraient nuire à l'expérience utilisateur.
Critère de complétion:	Les applications ont un temps d'attente très minime ou imperceptible.
Considérations spéciales:	Tous les étudiants de l'équipe 110 feront les tests afin de faire le plus de trafic au serveur possible durant un laps de temps.

3.1.5. Tests de charge

Objectif de test:	Vérifier que les dessins ainsi que son effacement (par trait ou par point) soit performant, soit pour le client local ou pour les clients participants dans le même jeu.
Technique:	Le testeur doit faire des extrêmement longs trait et ensuite tenter d'effacer rapidement avec les traits effacés à trait ou à point. Il doit vérifier que les effacements sont optimisés afin de faire des effacements sur un très grand nombre de petits traits.
Critère de complétion:	L'effacement est instantané et celui-ci ne fait pas "gelé" les applications lors de son fonctionnement.
Considérations spéciales:	

3.1.6. Tests de stress

Objectif de test:	Vérifier que le serveur ne s'arrête pas si plusieurs parties sont gardées sur la mémoire vive pour les dessins des joueurs virtuels.
Technique:	Les testeurs se créent chacun une parties solo pour la recherche de plusieurs dessins sur le serveur. Ils vérifient que le serveur performe comme s'il était le testeur est le seul à jouer en ce moment même.
Critère de complétion:	Le serveur ne doit pas s'arrêter et la latence des applications doit être à un temps normal.
Considérations spéciales:	

3.1.7. Tests de volume

Objectif de test:	Vérifier que les applications peuvent soutenir des larges sommes de données et que ceux-ci ne vont pas mettre les applications en panne.
Technique:	Le testeur doit envoyer extrêmement beaucoup de messages, de toute taille, dans les canaux de discussions. Ensuite, il se déconnecte et revient pour la recherche des messages envoyés précédemment. Il vérifie alors que le chargement de tous ces messages ne gèle pas l'application ni le serveur.
Critère de complétion:	Le chargement de messages doit être aussi rapide qu'avec quelques messages ou beaucoup de messages.
Considérations spéciales:	

3.1.8. Tests d'échec/récupération

Objectif de test:	Vérifier qu'un utilisateur est indépendant des autres, soit qu'un utilisateur ne peut pas ruiner une partie ou de logiciel d'autrui en général s'il a soudainement une défaillance.
Technique:	<p>Des testeurs commencent une ou plusieurs parties multijoueurs. Un des testeurs va ensuite se déconnecter/fermer son application afin de vérifier si les parties ont encore un comportement habituelle en regardant si les affichages de l'interface est adéquat selon le "flow".</p> <p>Le testeur déconnecter se reconnecte et les autres testeurs ne devraient pas avoir de problème avec leurs parties non plus.</p>
Critère de complétion:	Il n'y a pas de "crash" du serveur ou des logiciels autres clients lorsqu'un testeur a une "défaillance" de son logiciel.
Considérations spéciales:	

3.2. Outils

Les outils suivants seront utilisés au sein de la discipline de test:

Type de test	Outil
Tests de fonction	Windows 10, tablette/emulator
Tests d'interface usager	Windows 10, tablette/emulator
Tests d'intégrité des données	Windows 10, tablette/emulator, DataGrip, MongoDB Atlas
Tests de performance	Windows 10, tablette/emulator
Tests de charge	Windows 10, tablette/emulator
Tests de stress	Windows 10, tablette/emulator
Tests de volume	Windows 10, tablette/emulator
Tests d'échec/récupération	Windows 10, tablette/emulator

4. Ressources

4.1. Équipe de test

Rôle	Membre de l'équipe	Responsabilités
Testeur client lourd et concepteur des cas de tests	Pablo	Concepteur des cas de tests du client lourd
Concepteur des tests client lourd	François Xavier	Implémentation des testes du client lourd
Testeur client lourd	Duc-Thien	Exécutions des tests du client lourd
Testeur client léger et concepteur des cas de tests	Jeremy	Concepteur des cas de tests du client léger
Concepteur des tests client léger	Jacob	Implémentation des testes du client léger
Testeur client léger	Arthur	Exécutions des tests du client léger

4.2. Système

Afin de pouvoir utiliser le client lourd, celui-ci est fait avec WPF, donc il faut rouler l'exécutable sur un ordi avec Windows 10. Pour le client léger, c'est une application Android, il faut l'installer sur une tablette Galaxy Tab A 2019. Pour le serveur, celui-ci est déployé sur Heroku. Il faut réveiller le serveur préalablement puisqu'il est mis en sommeil après un certain temps sans activités.

5. Jalons du projet

Jalon	Effort	Date de début	Date de fin
Conception des cas de tests	6	2020-03-20	2020-03-25
Implémentation des tests des clients	8	2020-03-25	2020-03-26
Exécution des tests de fonctions et d'interface usager	6	2020-03-25	2020-03-26
Exécution des tests d'intégrité des données, de performances, de charge, stress, volume et échec/récupération	2	2020-03-27	2020-03-28
Conception des cas de tests	6	2020-04-01	2020-04-06
Implémentation des tests des clients	8	2020-04-06	2020-04-07
Exécution des tests de fonctions et d'interface usager	6	2020-04-08	2020-04-09
Exécution des tests d'intégrité des données, de performances, de charge, stress, volume et échec/récupération	2	2020-04-10	2020-04-11
Exécution des tests de fonctions et d'interface usager	8	2020-04-11	2020-04-13