
Équipe 102

PolyScrabble
Plan de tests logiciels

Version 1.3

Historique des révisions

Date	Version	Description	Auteur
2022-11-16	1.0	Choix de type de tests et écriture de ces tests	Marc-Antoine Baillargeon/Aghiles Gasselin
2022-11-24	1.1	Fin d'écriture de la partie stratégie de test	Marc-Antoine Baillargeon/Aghiles Gasselin
2022-11-28	1.2	Ajout tests de fonctions dans stratégie de test	Marc-Antoine Baillargeon/Aghiles Gasselin
2022-12-02	1.3	Formatage final	Aghiles Gasselin

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 de performance	6
3.1.4. Tests de charge	6
3.1.5. Tests d'échec/récupération	6
3.2. Outils	7
4. Ressources	7
4.1. Équipe de test	7
4.2. Système	8
5. Jalons du projet	8

Plan de tests logiciels

1. Introduction

Ce document a pour but de présenter les tests logiciels qui seront effectués sur notre application. Ces tests se basent sur notre SRS et vérifieront que notre application respecte celui-ci. Ils ont donc pour but de vérifier que notre application respecte ce que nous avons promis dans le SRS et sont une preuve du bon fonctionnement de notre application. Ce document fait partie des fichiers qui serviront d'assurance-qualité.

Pour commencer, le document énumère les exigences qui seront testées. Par la suite les tests choisis le test d'interface usager, le test de performance, le test de charge et finalement le test d'échec/récupération. Les outils utilisés pour tester ces exigences sont présentés par la suite les ressources utilisées pour effectuer ces tests.

2. Exigences à tester

Exigences	Tests
3.1 Mode de jeu	Tests de fonction, Tests d'interface usager
3.1.1 Mode de jeu classique	Tests de fonction, Tests d'interface usager, Tests de performance, Tests de charge, Tests d'échec récupération
3.1.2 Mode de jeu carte de pouvoir configurable	Tests de fonction, Tests d'interface usager, Tests de performance, Tests de charge, Tests d'échec récupération
3.2 Visibilité des parties	Tests de fonction, Tests d'interface usager, Tests de performance, Tests d'échec récupération
3.3 Interaction avec le plateau	Tests de fonction, Tests d'interface usager,

	Tests de performance
3.6 Résultat de fin de partie	Tests de fonction, Tests d'interface usager, Tests d'échec récupération
3.7 Matches favoris	Tests de fonction, Tests d'interface usager, Tests d'échec récupération
3.8 Comptes Utilisateurs	Tests de fonction, Tests d'interface usager, Tests de performance, Tests de charge, Tests d'échec récupération
3.9 Avatar	Tests de fonction, Tests d'interface usager, Tests d'échec récupération
3.10 Clavardage	Tests de fonction, Tests d'interface usager, Tests de performance, Tests de charge, Tests d'échec récupération
3.11 Configuration de l'application	Tests de fonction, Tests d'interface usager, Tests de performance, Tests d'échec récupération
3.13 Création de cookies	Tests de fonction, Tests d'échec récupération
3.15 Système de classement	Tests de fonction, Tests d'interface usager, Tests d'échec récupération
3.16 Actions provoque du son	Tests de fonction, Tests d'interface usager, Tests de performance, Tests de charge
4.1 Utilisabilité	Tests de fonction, Tests d'interface usager, Tests de performance
4.2 Fiabilité	Tests de fonction, Tests de performance
4.3 Performance	Tests de fonction, Tests de performance

3. Stratégie de test

3.1. Types de test

3.1.1. Tests de fonction

Objectif de test:	Vérifier que la fonction suit les exigences du srs remis.
Technique:	Le testeur doit avoir le srs sous les yeux et tester chaque point de celui-ci.
Critère de complétion:	Le testeur a observé que toutes les exigences du srs sont implémenter et fonctionnelles.
Considérations spéciales:	L'utilisateur est un humain adulte avec une bonne connaissance du français et une connaissance adéquate de la technologie.

3.1.2. Tests d'interface usager

Objectif de test:	Vérifier que l'utilisateur peut accéder à chaque pages auxquelles il aurait normalement accès. Vérifier que les boutons redirigent aux bonnes pages lorsqu'ils sont appuyés.
Technique:	Le testeur a une liste de pages et doit pouvoir accéder à chacune d'entre elles sans avoir besoin d'aide externe.
Critère de complétion:	Le testeur a réussi à accéder à chacune des pages tant qu'il a les droits d'accès.
Considérations spéciales:	L'utilisateur est un humain adulte avec une bonne connaissance du français et une connaissance adéquate de la technologie.

3.1.3. Tests de performance

Objectif de test:	Vérifier que les résultats des actions du client se produisent en moins de 500ms.
Technique:	Effectuer chaque action possible par un utilisateur et vérifier que le délai entre l'action et la conséquence.
Critère de complétion:	Le délai entre l'action et son résultat est inférieur à 500 ms.
Considérations spéciales:	Le temps de ping aux serveurs est inférieur à 100 ms.

3.1.4. Tests de charge

Objectif de test:	Tester que tous les clients sont synchronisés au niveau des actions effectuées (création de partie/envoi de message/placer/échanger des lettres).
Technique:	Prendre 4 personnes, les mettre à la page de la feature testées et envoyer en même temps pour ensuite envoyer des actions synchronisées.
Critère de complétion:	Si chaque client a reçu toutes les informations d'actions dans le bon ordre alors le test passe.
Considérations spéciales:	L'utilisateur est un humain adulte avec une bonne connaissance du français et une connaissance adéquate de la technologie.

3.1.5. Tests d'échec/récupération

Objectif de test:	Vérifier que les configurations de chaque utilisateur sont persistantes même en cas de crash.
Technique:	Changer tous les champs modifiables d'un profil utilisateur, faire crasher l'application (l'éteindre ainsi qu' éteindre le serveur), rallumer le tout et observer.
Critère de complétion:	Si chaque paramètre est le même que ceux modifiés précédemment le test passe.

Considérations spéciales:	Les champs qui doivent rester persistant sont seulement les champs dans la page de profil de l'utilisateur à savoir : le nom d'utilisateur, son avatar, son choix de thème par défaut, sa langue par défaut.
------------------------------	--

3.2. Outils

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

Type de test	Outil
Test de fonction	Tablette Android, Application électron sur Windows 10, Serveur déployé sur AWS.
Test d'interface usager	Tablette Android, Application électron sur Windows 10
Test de charge	Serveur déployé sur Aws et client sur electron + client sur Android 9.0
Test d'échec/récupération	Serveur déployé sur Aws et client sur electron + client sur Android 9.0
Test de Performance	Tablette Android, Application électron sur Windows 10 et la fonction console.time

4. Ressources

4.1. Équipe de test

Rôle	Membre de l'équipe	Responsabilités
------	--------------------	-----------------

Testeur des tests d'interface usager, de performance et de fonction	Marc-Antoine Baillargeon	Rédaction des tests/implémentations/exécutions et rédaction des résultats.
Testeur des tests de charge et d'échec/récupération	Aghiles Gasselin	Rédaction des tests/implémentations/exécutions et rédaction des résultats.
Testeur de tests de charge	Corentin Glaus	Exécution des tests
Testeur de tests de fonctions	Maximiliano Falicoff	Exécution des tests
Testeur de tests de charge et de fonctions	Stephane Toyo Demanou	Exécution des tests
Testeur d'interface usager	Mohamed Fenjiro	Exécution des tests

4.2. Système

Pour que chacun de nos tests soient valides, le serveur doit être déployé sur AWS dans une configuration identique à celle lors de la remise au client le 2 décembre. De plus, les clients utilisés doivent être déployés sur Electron (avec exécutable .exe) pour le client utilisant l'application sur Windows et en tant qu'un exécutable .apk pour le client utilisant une tablette android 9.0.

5. Jalons du projet

Jalon	Effort	Date de début	Date de fin
Planification des tests	2	17-11-2022	20-11-2022
Implémentations des tests	6	20-11-2022	25-11-2022
Exécution des tests	1	25-11-2022	25-11-2022
Résultats-Analyse des tests	5	30-11-2022	30-11-2022