
Équipe 112

Fais-moi un dessin
Plan de tests logiciels
Version 1.3

Historique des révisions

| Date | Version | Description | Auteur |
|------------|---------|-----------------------------------------------------------------------------------------------------------|-------------------------------|
| 2020-03-24 | 1.0 | Introduction, une partie des exigences à tester, ressources, outils, jalons et 50% des stratégies de test | Huyen Trang Dinh |
| 2020-04-01 | 1.1 | Retrait des tests de sécurité et contrôle d'accès et de charge | Georges Parfait Djimefo Kapen |
| 2020-04-07 | 1.2 | Rédaction de la partie 2. | Georges Parfait Djimefo Kapen |
| 2020-04-12 | 1.3 | Vérification du document final en conformité avec les exigences | Georges Parfait Djimefo Kapen |

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 stress | 6 |
| 3.1.6. Tests de volume | 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

Le logiciel que nous avons développé se doit d'être testé et évalué afin d'en assurer la qualité. Avant de procéder aux tests, nous étayons dans ce document les:

- Exigences à tester;
- Stratégies de tests à effectuer pour chaque exigence.

Ce document clarifie donc parallèlement nos attentes en termes de qualité pour chaque fonctionnalité développée. Nous détaillons aussi toutes ressources utilisées pour les tests logiciels, ainsi que les jalons du projet reliés à la discipline des tests.

2. Exigences à tester

| Exigences | Tests associés |
|------------------------------------------------|----------------------------|
| Cadriciel Poly Paint (Mode de jeu généralités) | Tests de fonction |
| Clavardage et canaux | Tests de fonction |
| Profil utilisateur (historique) | Tests de fonction |
| Profil utilisateur (historique détaillé) | Tests de fonction |
| Profil utilisateur (historique semi-détaillé) | Tests de fonction |
| Mode de jeu free-for-all | Tests de fonction |
| Mode de jeu sprint solo | Tests de fonction |
| Mode de jeu sprint coopératif | Tests de fonction |
| Mode de jeu cross-platform | Tests de fonction |
| Personnalité des joueurs virtuels | Tests de fonction |
| Profil utilisateur (avatar) | Tests de fonction |
| Création de jeu (Manuelle I, II) | Tests de fonction |
| Création de jeu (Assistée I) | Tests de fonction |
| Création de jeu (Assistée II) | Tests de fonction |
| Création de jeu (Assistée III) | Tests de fonction |
| Effets visuels et sonores | Tests de fonction |
| Tutoriel (non interactif) | Tests de fonction |
| Interfaces Usagers | Tests d'interfaces usagers |

| | |
|---------------------------------|----------------------------------|
| Protection de données sensibles | Tests d'intégrité des données |
| Performance | Tests de performance |
| Contraintes générales | Tests de stress, tests de volume |

3. Stratégie de test

Cette section décrit les stratégies de tests qui seront utilisées pour tester le logiciel *Fais-moi un dessin*. Les types de tests seront exposés sous forme de tableaux, montrant leur objectif, technique, critère de complétion et autres considérations.

3.1. Types de test

3.1.1. Tests de fonction

| | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objectif de test: | Vérifier que les fonctionnalités sont implémentées correctement selon le SRS et les cas d'utilisation. |
| Technique: | Les tests sont faits en boîte noire. Le testeur doit exécuter la fonctionnalité, puis comparer les résultats avec le cas d'utilisation qui y est associé. |
| Critère de complétion: | Les résultats des tests remplissent les exigences du SRS et concordent avec les cas d'utilisation. |
| Considérations spéciales: | |

3.1.2. Tests d'interface usager

| | |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objectif de test: | Vérifier que l'utilisateur peut interagir avec toutes les interfaces qui lui sont disponibles. Vérifier que l'utilisateur peut naviguer à travers toutes les vues qui lui sont disponibles. |
| Technique: | Différents testeurs externes doivent tenter de naviguer entre toutes les vues possibles. Ils doivent aussi tester les boutons et saisir du texte valide et invalide dans tous les champs de texte. |
| Critère de complétion: | Les champs de textes et boutons ont des comportements attendus selon les permissions de l'utilisateur. La navigation se fait adéquatement selon les permissions de l'utilisateur. |
| Considérations spéciales: | |

3.1.3. Tests d'intégrité des données

| | |
|---------------------------|--------------------------------------------------------------------------------------------------|
| Objectif de test: | Vérifier que la base de données ne contient jamais des données invalides. |
| Technique: | Le testeur doit tenter, à travers des requêtes HTTP, d'envoyer des données valides et invalides. |
| Critère de complétion: | La base de données ne contient que des données valides. |
| Considérations spéciales: | |

3.1.4. Tests de performance

| | |
|---------------------------|-----------------------------------------------------------------------------------------------------------|
| Objectif de test: | Vérifier que les temps de réponse du serveur sont en accord avec ceux prescrits par le SRS. |
| Technique: | Le testeur exécute des échanges entre les clients et le serveur et recueille les temps de réponse moyens. |
| Critère de complétion: | Les temps de réponse moyens respectent les temps prescrits par le SRS. |
| Considérations spéciales: | |

3.1.5. Tests de stress

| | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objectif de test: | Vérifier que le logiciel effectue ses opérations sans erreurs lorsque soumis à une grande quantité d'opérations en cours. |
| Technique: | Le testeur crée une quantité élevée de requêtes sur le logiciel via multiples clients différents. Ensuite, il vérifie que les données sont restées cohérentes à travers tous les clients et la base de données. |
| Critère de complétion: | Il n'y a pas d'erreur liée à la concurrence et l'exécution. Les données demeurent intègres. |
| Considérations spéciales: | |

3.1.6. Tests de volume

| | |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objectif de test: | Vérifier que le logiciel est en mesure de supporter un grand nombre de données sans limiter le serveur dans ses opérations. |
| Technique: | Le testeur ajoute une grande quantité de données dans la base de données et vérifie qu'il n'y a pas de panne qui se produit. Ensuite, le testeur vérifie que l'accès à la base de données du logiciel ne cause pas de problème. |
| Critère de complétion: | Aucune panne n'est causée suite aux tests et toutes les opérations serveur sont complétées de manière adéquate. |
| Considérations spéciales: | |

3.2. Outils

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

| Type de test | Outil |
|-------------------------------|-----------------------------------------------------|
| Tests de performance | Tablette Android, PC (Win10), Serveur (Node) |
| Tests d'intégrité des données | Serveur (Node), MongoDB |
| Tests de fonction | Tablette Android, PC (Win10) |
| Tests d'interface usager | Tablette Android, PC (Win10), Utilisateurs externes |
| Tests de volume | Tablette Android, PC (Win10), MongoDB |
| Tests de stress | Tablette Android, PC (Win10), MongoDB |

4. Ressources

Cette section concerne les ressources humaines et matérielles variées qui seront utilisées lors des étapes relatives à la discipline de test.

4.1. Équipe de test

| Rôle | Membre de l'équipe | Responsabilités |
|---------------------------------------------------|-------------------------------|-----------------------------------------------------------------------------------------------|
| Testeur exigences non fonctionnelles client lourd | Georges-Parfait Djimefo-Kapen | Exécuter les tests sur les exigences non fonctionnelles sur le client lourd |
| Testeur client lourd | Huyen Trang Dinh | Exécuter les tests sur les exigences fonctionnelles sur le client lourd |
| Concepteur de tests client lourd | Ismael Gbian | Écrire les tests pour les exigences fonctionnelles et non fonctionnelles pour le client lourd |
| Testeur exigences non fonctionnelles client léger | Roger Kazma | Exécuter les tests sur les exigences non fonctionnelles sur le client léger |
| Concepteur de tests client léger | Ibrahim Choukier | Écrire les tests pour les exigences fonctionnelles et non fonctionnelles pour le client léger |

| | | |
|----------------------|---------------------------|--------------------------------------------------------------------------------------|
| Testeur client léger | Talet Kayhan | Exécuter les tests sur les exigences fonctionnelles sur le client léger |
| Testeur UX externe | Ressource anonyme externe | Exécuter des tests sur les exigences non fonctionnelles et l'expérience utilisateur. |

4.2. Système

Les tests seront exécutés sur des tablettes et des ordinateurs. La quantité d'ordinateurs utilisée dans les tests peut varier entre 0 et 3, tandis que la tablette, entre 0 et 2. Les ordinateurs devront avoir au moins 4Gb de mémoire vive, ainsi qu'une résolution minimale de 1080x1920 sur Windows 10. Les tablettes sont des Samsung Galaxy Tab A, roulant Android. Enfin, les tests sont effectués sur un serveur Heroku. La base de données à tester est gérée par MongoDB.

5. Jalons du projet

| Jalon | Effort | Date de début | Date de fin |
|--------------------------------------------|--------|---------------|-------------|
| Définition des cas de test | 2 | 2020-03-23 | 2020-03-30 |
| Écriture des tests de fonction | 1 | 2020-03-23 | 2020-03-30 |
| Écriture des tests d'intégrité des données | 1 | 2020-03-23 | 2020-03-30 |
| Écriture des tests d'interface usager | 1 | 2020-03-30 | 2020-04-06 |
| Écriture des tests de stress | 1 | 2020-03-30 | 2020-04-06 |
| Écriture des tests de volume | 1 | 2020-03-30 | 2020-04-06 |
| Écriture des tests de performance | 1 | 2020-04-06 | 2020-04-13 |
| Exécution des tests | 1 | 2020-04-06 | 2020-04-13 |
| Analyse des résultats | 2 | 2020-04-06 | 2020-04-13 |