

Air Instruments
Plan de tests logiciels
Version 2.0

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

Historique des révisions

Date	Version	Description	Auteur
2013-02-12	1.0	Rédaction du plan de tests.	Alexandre
2014-02-14	1.1	Révision.	François
2014-04-07	2.0	Révision du plan de tests.	Félix

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

Table des matières

1.	Introduction	4
1.1	But	4
1.2	Mise en contexte	4
1.3	Documentation	4
2.	Exigences à tester	4
3.	Stratégie de test	6
3.1	Types de test	6
3.1.1	Tests de fonction	6
3.1.2	Tests d'interface usager	7
3.1.3	Tests de performance	7
3.1.4	Tests de charge	8
3.1.5	Tests de stress	8
3.1.6	Tests de volume	9
3.1.7	Tests de sécurité et de contrôle d'accès	9
3.1.8	Tests d'échec/récupération	9
3.2	Outils	10
4.	Ressources	10
4.1	Équipe de test	10
4.2	Système	11
5.	Jalons du projet	11

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

Plan de tests logiciels

1. Introduction

1.1 But

Ce plan de tests pour le *Air Instruments* a pour but d'identifier les composantes à tester, d'explicitier les stratégies de tests utilisées pour les tester ainsi que les ressources utilisées pour y arriver, et enfin de détailler les jalons relatifs à la discipline de tests.

1.2 Mise en contexte

Dans le cadre du projet intégrateur de quatrième année, nous développons un logiciel qui permet de jouer des instruments de musique sans aucun support physique à l'aide de senseurs tel la Kinect. Comme le logiciel sera présenté à une compétition, la stabilité est très importante et nous prévoyons donc nous concentrer sur la minimisation du nombre de bogues.

1.3 Documentation

Les documents suivants étaient disponibles pour l'identification, l'exécution et la validation des tests:

- SRS
- Document d'architecture (cas d'utilisation, diagrammes de séquences)

2. Exigences à tester

Fonctionnalité	Interface	Calcul
2.5.1 L'utilisateur aura à sa disposition une interface pour choisir le mode de jeu.	S'assurer qu'il est possible de choisir le mode de jeu et que ce choix est clair à l'utilisateur.	S'assurer que les calculs faits pour détecter les gestes de changement de mode de jeu sont robustes et fonctionnent avec divers utilisateurs.
2.5.2 L'utilisateur peut accéder au mode libre.	S'assurer qu'il est possible d'accéder au mode libre.	Aucun
2.5.3-2.5.5 L'utilisateur peut accéder au mode assisté du piano, de la guitare et de la batterie.	S'assurer qu'il est possible d'accéder au mode assisté.	Aucun
2.6.* Le présentateur pourra contrôler le logiciel de différentes façon afin de faciliter la présentation.	S'assurer que les opérations spécifiées dans le SRS sont disponibles et fonctionnelles.	Aucun
2.7.1 L'utilisateur devra pouvoir choisir un instrument facilement avec l'aide de gestes.	S'assurer l'interface guidant l'utilisateur dans le choix de l'instrument est clair et que l'instrument se positionne correctement par la suite.	Calcul de la position de l'instrument à partir des données reçues par les senseurs.
2.7.2 Du guidage sera affiché à l'utilisateur pour le choix de l'instrument.	Des pictogrammes indiquent les gestes à faire pour choisir chaque instrument. L'interface indique lorsqu'un geste est en cours de détection.	Aucun
2.8.1 L'utilisateur devra pouvoir choisir l'instrument « piano ».	S'assurer qu'il est possible de choisir de jouer du piano.	Aucun

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

2.8.4 L'utilisateur pourra jouer sur jusqu'à 3 octaves du piano.	S'assurer que le piano affiché est l'écran possède au moins 5 octaves.	Aucun
2.8.5 La grosseur des notes du piano sera suffisante pour assurer que l'utilisateur ait de la facilité à atteindre les notes.	S'assurer que la grosseur des notes affichées à l'écran est considérée par un grand nombre d'utilisateurs comme : <ul style="list-style-type: none"> • Réaliste • Suffisante pour ne pas avoir de la difficulté à positionner un doigt sur une note précise. 	Aucun
2.9.1 Il sera possible de choisir la batterie comme instrument jouable	S'assurer qu'il existe un geste permettant de choisir la batterie comme instrument.	Calcul des percussions à l'aide des articulations fournies par la Kinect.
2.9.1 Une vue à la première personne sera affichée présentant la position des doigts ou des baguettes au-dessus de l'instrument pour le piano et la batterie.	S'assurer que pour le piano et la guitare l'instrument est affiché ainsi que les baguettes ou les mains selon l'instrument.	Aucun
2.9.2 Tous les instruments à percussion de la batterie seront jouables avec des gestes simulant leur jeu respectif.	S'assurer qu'il existe des gestes pour jouer chaque instrument à percussion d'une batterie.	Calcul des positions des baguettes à l'aide des articulations des bras.
2.9.4 La pédale pourra être utilisée pour simuler le jeu de la grosse caisse d'une batterie.	S'assurer qu'il existe un geste simulant le jeu d'une grosse caisse de la batterie et que celui-ci est considéré intuitif et représentatif de la réalité par un grand nombre d'utilisateurs.	Calcul de la position du pied à l'aide de l'articulation du genou.
2.10.1 Il sera possible de choisir la guitare comme instrument jouable.	S'assurer qu'il existe un geste qui permet de choisir la guitare comme instrument jouable.	Aucun
2.10.1 Une vue à la troisième personne présentant un personnage avec la guitare sera affichée lorsque l'instrument choisi est la guitare.	S'assurer que lorsque l'instrument choisi est la guitare un personnage avec une guitare soit affiché.	Aucun
2.10.2 Il sera possible de jouer de la guitare en déplaçant une main sur le manche de la guitare.	S'assurer que le geste de la main passant sur le manche de la guitare est reconnu.	Calcul de la note à jouer selon la position d'une des mains.
2.16.* Des tutoriaux aident l'utilisateur à apprivoiser les instruments.	Les instructions des tutoriaux sont affichées et lues à voix haute clairement.	Aucun

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

3. Stratégie de test

3.1 Types de test

3.1.1 Tests de fonction

Objectif de test:	S'assurer que tous les comportements observables décrits dans le SRS ont été implémentés. S'assurer que tous les cas d'utilisation décrits dans le document d'architecture peuvent être réalisés.
Technique :	<p>Pour chaque comportement observable décrit dans le SRS :</p> <ul style="list-style-type: none"> Observer le comportement dans l'application. S'assurer qu'il correspond exactement à la description du SRS. Lorsque c'est possible, tenter d'effectuer des actions invalides en lien avec le comportement observable. S'assurer que lors d'actions invalides, une réponse claire est offerte à l'utilisateur pour lui permettre de savoir ce qu'il doit corriger. S'assurer qu'il n'y a aucun «crash» ou comportement inattendu. <p>Pour chaque cas d'utilisation décrit dans le document d'architecture :</p> <ul style="list-style-type: none"> Réaliser le cas d'utilisation dans l'application.
Critère de complétion :	<ul style="list-style-type: none"> Tous les comportements observables décrits dans le SRS ont pu être observés et correspondent exactement à leur description (effectuer les corrections au besoin). Tous les cas d'utilisation peuvent être réalisés (effectuer les corrections au besoin).
Considérations spéciales:	<p>Les tests doivent être effectués par divers utilisateurs ayant des profils différents:</p> <ul style="list-style-type: none"> Membres de l'équipe et utilisateurs externes. Personnes de différentes tailles et différents âges. Personnes connaissant la musique ou non. <p>En effet, l'utilisation d'une interface «naturelle» nous met à risque qu'une exigence fonctionnant pour un utilisateur soient inutilisables pour d'autres.</p>

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

3.1.2 Tests d'interface usager

Objectif de test:	S'assurer que l'interface utilisateur est uniforme et respecte les règles d'utilisabilité enseignées dans le cours LOG2420.
Technique:	<p>Pour chaque écran de l'application, un testeur s'assure que :</p> <ul style="list-style-type: none"> • L'écran respecte les standards de navigation (titre, bouton retour, boutons vers les prochains écrans). • Des messages ou des indices visuels standards sont affichés pour toutes les actions invalides identifiées. • Les gestes à effectuer pour chaque comportement des requis sont clairement indiqués à l'utilisateur et faciles à faire. <p>Pour chaque écran de l'application, un testeur observe un néophyte tenter d'exécuter les comportements observables décrits dans le SRS et s'assure que :</p> <ul style="list-style-type: none"> • Les comportements observables peuvent être atteints dans des délais raisonnables spécifiés avant l'exécution des tests. • Le néophyte obtient les résultats voulus suite à ses actions. • Le néophyte s'assure que tous les gestes sont détectés de façon précise et dans un délai acceptable.
Critère de complétion:	Chaque écran respecte les standards fixés dans l'équipe. Un néophyte est capable d'utiliser l'application selon les attentes.
Considérations spéciales:	La participation d'un néophyte est nécessaire pour exécuter ces tests.

3.1.3 Tests de performance

Objectif de test:	S'assurer que les délais de latence pour le son et pour l'image sont assez bas pour assurer une expérience utilisateur fluide. S'assurer que le logiciel fonctionne dans des conditions sonores et de lumière variées. S'assurer que les données transférées ne dépassent pas la capacité des contrôleurs des ports utilisés par les périphériques.
Technique:	<p>Tester sur une machine standard avec des spécifications définies pour s'assurer que le logiciel s'exécute sans problèmes pour une utilisation normale.</p> <p>Tester sur une machine standard dans plusieurs environnements sonores et de lumière.</p> <p>Tester avec tous les périphériques connectés afin de vérifier que cela ne surcharge pas les contrôleurs de ports.</p>
Critère de complétion:	Dans n'importe quelle condition le délai de réponse pour le son et l'image est fluide. Le logiciel doit être résistant à des niveaux de sons raisonnables et doit fonctionner sous des luminosités relativement faibles ou élevées, avec parfois des ombres sur les parties du corps des joueurs.
Considérations spéciales:	

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

3.1.4 Tests de charge

Objectif de test:	S'assurer que les exigences non fonctionnelles du SRS ayant rapport à la performance soient comblées. S'assurer que dans les pires conditions le processeur suffise dans tous les cas à effectuer les calculs.
Technique:	<p>Processeur</p> <ul style="list-style-type: none"> • Effectuer des actions complexes qui nécessitent pour le logiciel d'effectuer beaucoup de calcul afin de tester les charges maximales imposées au processeur. • Faire passer plusieurs personnes devant la Kinect pour vérifier que cela n'affecte pas trop le temps de traitement. • Mettre plusieurs objets n'étant pas des mains à proximité de la caméra Creative d'Intel.
Critère de complétion:	<p>Dans tous les cas, le logiciel devra rester fluide.</p> <p>Les comportements indiqués dans le SRS devront être respectés</p> <p>La mémoire utilisé par une instance de jeu devra respecter les valeurs définies dans le SRS.</p>
Considérations spéciales:	

3.1.5 Tests de stress

Objectif de test:	Découvrir les problèmes liés à des ressources non libérées suite à une utilisation intensive.
Technique:	<p>Utiliser les fonctionnalités de l'application en changeant très fréquemment de mode. Observer l'utilisation de mémoire faite par les différents SDKs et l'application principale, dans le but de détecter des problèmes de gestion des ressources suite à très haute fréquence aux fonctions de traitement d'images.</p> <p>Coder des tests appelant les fonctions de traitement d'image à très haute fréquence à partir de différents threads en observant l'utilisation de mémoire.</p>
Critère de complétion:	La quantité de ressource utilisée par l'application doit plafonner même sous une utilisation intensive.
Considérations spéciales:	

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

3.1.6 Tests de volume

Objectif de test:	Vérifier que le logiciel fonctionne bien lorsque plusieurs personnes passent dans le champ de vision de la caméra et que les joueurs sont tout de même détectés.
Technique:	<ul style="list-style-type: none"> Faire passer plusieurs personnes devant la caméra jusqu'au point où cela cause des problèmes dans le logiciel.
Critère de complétion:	Il faut s'assurer qu'un nombre minimal d'interférence ou de bruit en arrière-plan ne vienne pas perturber l'expérience de jeu.
Considérations spéciales:	

3.1.7 Tests de sécurité et de contrôle d'accès

Objectif de test:	Vérifier que les joueurs n'ont pas accès au mode contrôleur et que l'accès ne leur est pas non plus visible.
Technique:	<ul style="list-style-type: none"> S'assurer que les contrôles administrateur peuvent seulement être effectués par les démonstrateurs (par le contrôle de l'accès aux périphériques utilisés ou par un contrôle d'accès dans le logiciel).
Critère de complétion:	Seuls les démonstrateurs ont accès aux options administrateurs.
Considérations spéciales:	

3.1.8 Tests d'échec/récupération

Objectif de test:	S'assurer qu'une défaillance temporaire au niveau du calcul des articulations ne met pas en péril la validité des données subséquentes. S'assurer que le jeu peut récupérer d'une quelconque défaillance temporaire dans les algorithmes de calcul.
Technique:	<p>Tester des cas où la détection par la caméra cause des erreurs temporaires et vérifier que les algorithmes de calcul des positions détectent ces erreurs et que cela n'a pas d'influence sur le jeu.</p> <p>Provoquer volontairement un plantage de l'application et s'assurer qu'il est possible de la redémarrer rapidement.</p>
Critère de complétion:	Les défaillances temporaires induites par les algorithmes ou par les capteurs n'ont pas d'influence sur l'expérience de jeu.
Considérations spéciales:	

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

3.2 Outils

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

Type de test	Outil
Test de charge	Task Manager, SRS
Tests d'interface usager	SRS
Tests de performance	Lumières, haut-parleurs
Tests de stress	-
Tests de volume	Visual Studio
Tests de sécurité et de contrôle d'accès	Visual Studio
Tests d'échec/récupération	Visual Studio
Tests de fonction	-

4. Ressources

4.1 Équipe de test

Rôle	Membre de l'équipe	Responsabilités
Testeur en chef (piano, autres)	Alexandre	Planifier et distribuer les tests. Exécuter les tests fonctionnels pour le piano et les instruments optionnels.
Testeur (batterie, guitare)	Simon	Exécuter les tests non-fonctionnels pour la batterie et la guitare.
Testeur (batterie, guitare)	Félix	Exécuter les tests fonctionnels pour la batterie et la guitare.
Testeur (piano, autres)	François	Exécuter les tests non-fonctionnels pour le piano et les instruments optionnels.

Air Instruments	Version: 2.0
Plan de tests logiciels	Date: 2014-04-15

4.2 Système

Les tests du jeu devront être faits sur l'ordinateur portable de François car c'est celui qui sera utilisé pour la compétition.

- Processeur : Intel Core i7 – 4 coeurs @ 2.70 GHz
- Mémoire : 8 Go
- Carte graphique : NVIDIA Quadro K1000M et Intel HD Graphics 4000
- Espace disque : 500 Go
- Windows 7 ou 8

5. Jalons du projet

Jalon	Effort	Date de début	Date de fin
Plan de tests	5h	12/02/14	14/02/14
Design de tests	5h	15/02/14	15/03/14
Exécution de tests	10h	19/03/14	26/03/14
Évaluation de tests	5h	26/03/14	30/03/14