
Laval Virtual

Air Instruments
Plan de développement logiciel

Version 1.2

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

Historique des révisions

Date	Version	Description	Auteur
12/02/2014	1.0	Ébauche, mise en ligne	Félix
12/02/2014	1.1	Élaboration	Félix
12/02/2014	1.2	Correction de fautes d'orthographe	François

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

Table des matières

1.	Introduction	4
2.	Vue d'ensemble du projet	4
2.1	But du projet, portée et objectifs	4
2.2	Hypothèses et contraintes	4
2.3	Biens livrables du projet	5
3.	Organisation du projet	5
3.1	Structure d'organisation	5
3.2	Interfaces externes	5
3.3	Responsabilités	6
4.	Processus de gestion	6
4.1	Plan de projet	6
4.1.1	Planification des jalons	6
4.1.2	Objectifs d'itération	7
4.2	Suivi de projet et contrôle	10
4.2.1	Gestion des exigences	10
4.2.2	Contrôle de la qualité	10
4.2.3	Gestion de risque	11
4.2.4	Gestion de configuration	12

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

Plan de développement logiciel

1. Introduction

Le but de ce document est de décrire le projet Air Instruments effectué dans le cadre du concours Laval Virtual. Nous donnerons d'abord une vue d'ensemble du projet en traitant du but, de la portée, des objectifs visés, des hypothèses et contraintes, ainsi que des bien livrables. L'organisation du projet sera également traitée. Nous parlerons alors de la structure d'organisation de l'équipe, des interfaces avec les groupes externes et des responsabilités des membres de l'équipe. Finalement, le processus de gestion sera traité et nous présenterons notre plan de projet ainsi que nos méthodes de gestion pour effectuer un suivi et un contrôle adéquats.

2. Vue d'ensemble du projet

2.1 But du projet, portée et objectifs

Notre projet se conclut par une présentation de notre produit lors du concours étudiant Virtual Fantasy du 9 au 13 avril prochains. Notre but ultime est de gagner les 2 volets de ce concours et notre projet sera l'objet de la partie « Démonstrations ».

Nos objectifs principaux sont donc :

- Impressionner le jury
- Impressionner le grand public

Pour cela nous voulons avoir un produit

- Original
- Visuellement et auditivement accrocheur
- Adapté à l'environnement du concours
- Ayant un taux de panne et temps de réinitialisation très faibles
- Ayant un temps d'apprentissage très court
- Exempt de défauts apparents

2.2 Hypothèses et contraintes

Notre équipe est composée de 4 membres, et avons au moins 12 heures de réservations pour travailler en équipe dans un local commun par semaine. Pour le matériel, nous avons déjà accès à 4 Kinect de première génération et une caméra Senz3D de Creative avec kit de développement d'Intel. Nous espérons également avoir accès à une Kinect de deuxième génération.

La démonstration est faite à partir du 9 avril et nous partons en France le 4 avril. Nous devons donc avoir un produit finalisé pour la fin mars. Sur place l'environnement possède certaines caractéristiques contraignantes également. Nous devrons faire la présentation dans un espace de 2m x 2m, avec un éclairage possiblement variant et imprévisible ainsi que du bruit ambiant.

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

2.3 Biens livrables du projet

L'artéfact suivant devra être complété avant le 10 mars :

- Version préliminaire jouable et testable pour effectuer des « play tests » durant 3 semaines

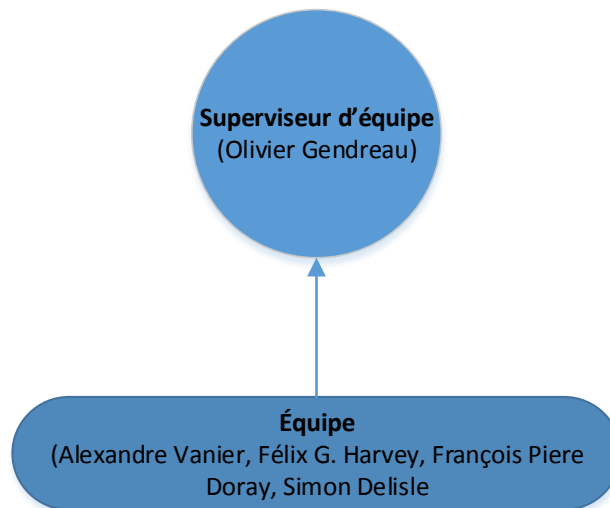
Les artéfacts suivant devront être complétés et livrés pour le 1^{er} avril :

- Exécutable du jeu accompagné des librairies dynamiques nécessaires.
- Affiche promotionnelle pour la démonstration
- Document présentant les résultats de tests, accompagné du plan de tests
- Vidéo promotionnel pour la démonstration (**optionnel**)

3. Organisation du projet

3.1 Structure d'organisation

Nous adoptons une structure latérale d'organisation, sans hiérarchie globales. La dynamique change en fonction des domaines, étant donné nos responsabilités et expertises différentes.



3.2 Interfaces externes

- Laval Virtual
 - Jérémy Pluinage : jpluinage@laval-virtual.org
- École Polytechnique de Montréal
 - Olivier Gendreau : olivier.gendreau@polymtl.ca
 - Benoit Forget : benoit.forget@hotmail.com
 - Alexandra Labbé : tresorier@aep.polymtl.ca
- Équipe 2013
 - Patrice Voutsinas : patrice.voutsinas@gmail.com
 - Guillaume Riendeau : guillaume.riendeau@gmail.com

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

3.3 Responsabilités

Simon Delisle

- Produit (configuration)
 - Responsable du contrôle des versions et de la traçabilité des exigences.
- Coordination/planification
 - Assurer le respect de l'échéancier
 - Choix de l'outil de planification et du serveur GIT (Redmine, githost, GitHub, BitBucket, ou autre)
 - Mise à jour des jalons et de l'échéancier
 - Division/attribution des tâches via l'outil de planification choisi

François Pierre Doray

- Technique (Outils)
 - Expert technique
 - Acquisition de connaissances sur les outils et technologies utilisables
 - S'assure du développement d'une architecture logicielle selon les règles de l'art

Alexandre Vanier

- Assurance qualité
 - Tests (de régression, de jouabilité, et autres au besoin)
 - Application des notions d'INF8301

Félix Gingras Harvey

- Représentant d'équipe
 - Communication avec les parties externes (école, professeur, représentants de Laval Virtual, etc.)
 - Rédactions de documents (demandes de financements, requêtes d'informations et autres)
- Processus/méthodologie
 - Assurer le respect du processus
 - Cibler et régler les faiblesses de la méthodologie utilisée

4. Processus de gestion

4.1 Plan de projet

4.1.1 Planification des jalons

1) Prototypage du piano et de la batterie : 3 février 2014

- Prototypes de piano et batterie fonctionnels
- Recherches effectuées sur les différents capteurs 3D
- Solution choisie pour le fonctionnement de la guitare

2) Instruments avancés : 24 février 2014

- Prototypes de piano et batterie avancés
- Prototype de guitare fonctionnel
- Choix finaux effectués sur les technologies utilisées

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

- Choix finaux effectués sur des éléments influençant la logistique et le « *gameplay* »
 - Nombre de joueurs
 - Nombre de rendus effectués
 - Nombre de capteurs utilisés

3) Air Instruments bêta : 10 mars 2014

- Instruments fonctionnels et raffinés
- Ajout de fonctionnalités « hors-instruments »
 - Choix du mode de jeu (mode assisté fonctionnel)
 - Choix de l'instrument
 - Contrôle « administrateur » (réinitialisation du squelette, etc.)

4) Air Instruments 1.0 : 1^{er} avril 2014

- Jeu testé intensivement (tests « portes-ouvertes », tests de régression, etc.)
- Failles corrigées
- Améliorations et ajustements finaux des instruments
- Amélioration et ajustement finaux du « *gameplay* »
- Améliorations visuelles (**optionnel**)

4.1.2 Objectifs d'itération

Tableau 1: Itérations du produit et objectifs

Itérations	Objectifs
12 janvier au 18 janvier (1 sem.)	Mettre en place du répertoire Github Élaborer le document de vision
19 janvier au 25 janvier (1 sem.) Prototypage du piano et de la batterie	Fournir des « <i>assets</i> » pour le piano et la batterie <ul style="list-style-type: none"> • Trouver/Enregistrer des sons • Fournir des modèles 3D Débuter le prototypage du piano et de la batterie <ul style="list-style-type: none"> • Mettre en place des scènes Unity avec sons Fournir une librairie de communication avec la Kinect à partir de Unity (SensorLib.dll)
26 janvier au 1 ^{er} février (1 sem.) Prototypage du piano et de la batterie	SensorLib <ul style="list-style-type: none"> • Conception d'algorithmes de reconnaissance de main

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

	<ul style="list-style-type: none"> • Accès aux données du squelette <p>Piano (Unity)</p> <ul style="list-style-type: none"> • Accéder aux données de SensorLib et jouer les notes selon ces données • Afficher l'image de profondeur <p>Batterie (Unity)</p> <ul style="list-style-type: none"> • Animer le squelette selon les informations fournies par SensorLib • Lier les collisions aux sons de la batterie
<p>2 février au 8 février (1 sem.)</p> <p>Instruments avancés</p>	<p>SensorLib</p> <ul style="list-style-type: none"> • Améliorer la reconnaissance de mains • Fournir les informations des joints des mains <p>Piano (Unity)</p> <ul style="list-style-type: none"> • Modéliser les joints des mains • Lier sons-collisions • Jouer du piano avec les collisions Unity <p>Batterie</p> <ul style="list-style-type: none"> • Prendre en considération des jambes (« <i>bass kick</i> » et « <i>high-hat</i> ») • Bloquer les mains et baguettes lors de collisions (sans traverser les instruments) <p>Faire fonctionner la caméra Senz3D d'Intel et utiliser le SDK avec la Kinect</p>
<p>9 février au 15 février (1 sem.)</p> <p>Instruments avancés</p>	<p>SensorLib</p> <ul style="list-style-type: none"> • Stabiliser les données envoyées • Améliorer les algorithmes concernant les mains <p>Batterie</p> <ul style="list-style-type: none"> • Améliorer le « <i>Face Tracking</i> » en utilisant les fonctionnalités de la Kinect <p>Guitare (Unity)</p> <ul style="list-style-type: none"> • Créer la scène Unity avec sons • Mouvement du squelette sans modèle de guitare • Modéliser la guitare <p>Piano (Unity)</p>

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

	<ul style="list-style-type: none"> Modéliser et animer des modèles de mains <p>Senz3D</p> <ul style="list-style-type: none"> Faire fonctionner le SDK d'Intel avec la caméra Kinect
16 février au 22 février (1 sem.) Instruments avancés	<p>Guitare</p> <ul style="list-style-type: none"> Implémenter la logique des notes jouées selon la tonalité, rythme, style, etc. <p>Batterie</p> <ul style="list-style-type: none"> Raffiner les algorithmes de logique pour augmenter le réalisme de la batterie Réduire la latence pour la batterie <p>Piano</p> <ul style="list-style-type: none"> Raffiner les algorithmes de logique pour augmenter le réalisme du piano Réduire la latence pour le piano Modéliser et animer des modèles de mains
23 février au 1 mars (1 sem.) Instruments avancés	<p>Modes de jeux</p> <ul style="list-style-type: none"> Combiner les scènes pour les 3 instruments Enregistrer/monter les chansons du mode assisté Implémenter la logique du mode assisté (choix du mode, choix de chanson) Implémenter le choix de l'instrument
2 mars au 8 mars (1 sem.) Instruments avancés	<p>Implémenter les contrôles administrateur</p> <ul style="list-style-type: none"> Réinitialiser de la scène Supprimer des faux squelettes captés <p>Améliorer la jouabilité des instruments</p> <ul style="list-style-type: none"> Cibler/Régler les problèmes les plus évidents
9 mars au 15 mars (1 sem.) Air instruments bêta	<p>Débuter les « <i>play tests</i> »</p> <p>Ajuster le jeu selon les problèmes rencontrés</p> <p>Améliorer la jouabilité des instruments (si les ressources le permettent)</p> <ul style="list-style-type: none"> Profiler l'application

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

	<ul style="list-style-type: none"> • Réduire la latence au maximum
16 mars au 22 mars (1 sem.) Air instruments bêta	Continuer les « <i>play tests</i> » Ajuster le jeu selon les problèmes rencontrés Améliorer la jouabilité des instruments (si les ressources le permettent) <ul style="list-style-type: none"> • Profiler l'application • Réduire la latence au maximum Implémenter les requis souhaitables/optionnels (si les ressources le permettent)
23 mars au 29 mars (1 sem.) Air instruments bêta	Continuer les « <i>play tests</i> » Ajuster le jeu selon les problèmes rencontrés Améliorer la jouabilité des instruments (si les ressources le permettent) <ul style="list-style-type: none"> • Profiler l'application • Réduire la latence au maximum Implémenter les requis souhaitables/optionnels (si les ressources le permettent)
30 mars au 5 avril (1 sem.) Air Instruments 1.0	Finaliser le produit Remise du livrable finale

4.2 Suivi de projet et contrôle

4.2.1 Gestion des exigences

Ce projet étant pour nous la première utilisation d'Unity, et le premier contact avec la réalité virtuel, il est évident que certaines exigences sont appelées à changer. Nous devons mettre en place des outils pour mesurer et contrôler ces changements.

- Document de spécifications des requis du système
 - Permet de recenser et numéroté les exigences à fins de références
- Document de traçabilité des exigences
 - Permet d'avoir un état actuel des exigences dans un format concis
- Tests de régressions
 - Permet de savoir si les changements affectent des exigences déjà complétées

4.2.2 Contrôle de la qualité

Nos objectifs principaux étant surtout orientés sur l'expérience utilisateur, le principal moyen de mesurer et augmenter la qualité de notre logiciel sera par les tests utilisateurs durant les 3 semaines dédiées aux « *play tests* ». Nous pourrons alors avoir plusieurs opinions sur notre produit et avoir un aperçu de l'appréciation des gens globalement. Les tests de régression nous seront utiles ici aussi afin de ne pas perdre de qualité durant le développement.

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

4.2.3 Gestion de risque

La description des risques suit la convention suivante :

- Ampleur : sur une échelle de 1 à 10, 10 étant le risque le plus élevé. Cette analyse est basée sur la probabilité d'occurrence du risque, ainsi que ses impacts.
- Description : une description textuelle du risque ainsi que les problèmes attendus.
- Impact : échelle définissant la portée du risque
 - C – critique (affecte le projet en entier)
 - E – élevé (affecte les fonctionnalités principales du système)
 - M – moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
 - F – faible (l'acceptation du risque est une stratégie envisageable)
- Facteurs : aspects (métriques) du système pouvant être compromis.
- Stratégie de gestion : mesures à prendre afin de gérer le risque.

1 – Utilisation de capteurs 3D				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Bien que la majorité des capteurs 3D fonctionnent de manières semblables, aucun membre de l'équipe n'a travaillé avec des technologies de ce genre auparavant et nous ne connaissons pas leurs limites.	C	Dates de fin de dév. de prototypes	Réduire l'incertitude en prototypant le plus tôt possible dans le cycle de développement.

2 – Utilisation d'Unity 3D				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Bien que le moteur 3D Unity facilite grandement le développement de jeux vidéo, il est nouveau pour tous les membres de l'équipe.	M	Dates de fin de dév. de prototypes et du jeu principal	Réduire l'incertitude en se familiarisant avec l'engin Unity même avant la phase de construction et d'élaboration.

3 – Concours en France				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Un concours à l'étranger implique un déplacement de matériel, ainsi qu'un environnement non contrôlé (passants, éclairage, etc.)	E	Vitesse de déploiement Facilité d'utilisation	Ici les tests de déploiement dans plusieurs conditions d'éclairages et possiblement plusieurs environnements physiques le plus tôt possible dans la session nous semblent être la meilleure solution.

Air Instruments	Version : 1.2
Plan de développement logiciel	Date : 2014-02-14

4.2.4 Gestion de configuration

Notre équipe utilisera un entrepôt fourni par Github.com pour appliquer et gérer les modifications. Le portail du site nous servira également d'outils de gestion, grâce à ses nombreuses fonctionnalités concernant les tâches ou problèmes (« *Issues* »), les jalons (« *Milestones* »), et autres.

Afin de pouvoir déterminer à chaque itération si nous avons atteint les objectifs fixés, des tâches ont été créées à l'avance. Ces tâches ont été associées à des exigences et à des itérations. Les membres de l'équipe sont tenus d'indiquer pour chacun de leurs commits à quelle tâche il s'applique et de mettre à jour le pourcentage d'avancement. Ainsi, nous pouvons savoir à tout moment quel pourcentage de l'itération en cours est complété. Nous nous assurons aussi qu'aucun requis n'est oublié.

Les artefacts du projet sont déjà classés dans des dossiers représentatifs comme « Documentations », « Prototypes » ou « AirInstruments » pour le jeu principal. Des dossiers seront créés pour les versions bêta et finale. Les changements seront automatiquement identifiés par les numéros de « *commits* », qui eux peuvent être liés aux « *issues* ».