Air Instruments

Spécifications des requis du système (SRS)

Version 1.0

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2014-01-15 | 1.0 | Ébauche du SRS | Alexandre Vanier |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. Introduction 4

1.1 But 4

1.2 Définitions, acronymes et abréviations 4

1.3 Vue d’ensemble du document 4

2. Description globale 4

2.1 Caractéristiques des usagers 4

2.2 Interfaces 4

2.2.1 Interfaces usagers 4

2.2.2 Interfaces matérielles 4

2.2.3 Interfaces logicielles 4

2.2.4 Interfaces de communication 4

2.3 Contraintes générales 5

2.4 Hypothèses et dépendances 5

3. Exigences fonctionnelles 5

3.1 <Exigence fonctionnelle 1> 5

4. Exigences non-fonctionnelles 5

4.1 Utilisabilité 5

4.1.1 <Exigence d’utilisabilité 1> 5

4.2 Fiabilité 5

4.2.1 <Requis de fiabilité 1> 6

4.3 Performance 6

4.3.1 <Requis de performance 1> 6

4.4 Maintenabilité 6

4.4.1 <Requis de maintenabilité 1> 6

4.5 Contraintes de conception 6

4.5.1 <Contrainte de conception 1> 6

4.6 Sécurité 6

4.7 Exigences de la documentation usager en ligne et du système d’assistance 6

4.8 Normes applicables 6

ANNEXE A: Glossaire 7

Spécifications des requis du système (SRS)

# Introduction

## But

Le SRS décrit le comportement externe d’une application. Il décrit aussi les exigences non fonctionnelles, les contraintes de conception, ainsi que les autres facteurs nécessaires à la description complète des exigences du logiciel à développer.

## Définitions, acronymes et abréviations

[Cette sous-section fournit la définition de tous les mots, acronymes et abréviations nécessaires à l’interprétation adéquate de ce SRS. L’information peut être fournie par référence à un glossaire.]

## Vue d’ensemble du document

[Cette sous-section décrit l’organisation du document.]

# Description globale

[Décrire brièvement en quoi consiste le logiciel à développer (typiquement de 3 à 5 lignes)]

Air Instruments est un logiciel à but récréatif permettant à l’utilisateur de jouer de la musique sans avoir besoin d’instruments physiques. Le logiciel détecte les mouvements faits par l’utilisateur à l’aide d’un capteur et effectue un traitement qui permet d’obtenir un son réaliste.

## Caractéristiques des usagers

[Définir les caractéristiques des usagers potentiels du logiciel.]

Air Instruments est un projet réalisé dans le cadre de la compétition Laval Virtual, il est donc adapté au grand public mais en visant plus spécifiquement les experts en réalité virtuelle qui jugeront le produit. Autrement dit, le logiciel doit être performant du point de vue technologique mais tout de même simple à utiliser pour un utilisateur lambda.

## Interfaces

### Interfaces usagers

[Décrire brièvement les interfaces usagers qui seront implémentées par le logiciel]

### Interfaces matérielles

[Définir les interfaces matérielles qui seront supportées par le logiciel]

Le logiciel utilisera un ou plusieurs capteurs de mouvement tel Kinect ainsi qu’un projecteur pour projeter les images traitées sur une toile. Il est aussi possible que des écrans supplémentaires soient utilisés pour séparer l’interface du joueur de ce qui est vu par les spectateurs. Il faudra aussi des haut-parleurs pour faire jouer le son des instruments.

### Interfaces logicielles

[Définir les interfaces logicielles du système logiciel. Il peut s’agir de composantes achetées, réutilisées, ou développées en parallèle au système.]

Le moteur Unity sera utilisé pour la partie qui sera affichée à l’écran (les modèles 3D et leur logique d’affichage). Un pilote et probablement un SDK seront utilisés pour gérer la capture d’images et pour avoir des fonctions de traitement de base.

### Interfaces de communication

[Décrire toute interface de communication avec d’autres systèmes ou dispositifs (LAN, sériel, parallèle, USB, etc.)]

La/les Kinect (ou autre) sera branchée à un ordinateur qui recevra les images et les traitera. La scène à afficher sera ensuite projetée à l’aide d’un projecteur et la vue en FPS pourra possiblement être transmise à des écrans supplémentaires installés en face du/des joueurs.

## Contraintes générales

[Détailler les contraintes générales, incluant les contraintes de mémoire relatives au logiciel à développer.]

L’espace disponible en compétition sera de 2m x 2m. Le logiciel doit fonctionner bien dans une pièce de ces dimensions. Le logiciel doit aussi fonctionner avec un niveau de bruit élevé et avec un éclairage variant.

## Hypothèses et dépendances

[Énoncer les hypothèses et dépendances, s’il y a lieu. Cette section peut être supprimée au besoin.]

# Exigences fonctionnelles

[Cette section décrit les exigences fonctionnelles dans un langage naturel, mais structuré. Cette section est typiquement organisée par groupe de fonctionnalités.

Le type (essentiel, souhaitable ou optionnel) de chaque exigence fonctionnelle doit être indiqué.]

## Instruments obligatoires

Le piano et la batterie devront être jouables.

**Essentiel**

## Instruments optionnels

La guitare, divers instruments à percussions et divers instruments à vent pourront être jouables.

**Souhaitable**

## Mode libre

Un mode libre sera disponible où l’utilisateur pourra jouer de l’instrument choisi. Dans ce mode, seul le son des instruments sera joué par les haut-parleurs.

**Essentiel**

## Mode assisté

Un mode assisté sera disponible. Dans ce mode, une musique pré-selectionnée sera jouée et sera mise en sourdine lorsque l’utilisateur jouera d’un instrument. Lorsque l’utilisateur arrête de jouer, la musique est à de nouveau jouée.

**Essentiel**

## Mode néophyte

Un mode néophyte sera disponible. Dans ce mode, l’utilisateur sera assisté par le logiciel dans la pratique de l’instrument choisi. Pour un piano, par exemple, les notes à jouer pourraient être mis en surbrillance à l’avance ou alors on pourrait fournir l’enchainement de notes à suivre dans une sorte de « tableau » à la Guitar Hero/Rock Band.

**Optionnel**

## Détection automatique des instruments

Les instruments devront être détectés automatiquement. Ainsi, lorsqu’un joueur effectue un mouvement similaire à celui de quelqu’un qui jouerait du piano, un piano apparaîtra à l’endroit où il joue afin de servir de référence. Un mécanisme similaire sera utilisé pour tous les instruments.

**Essentiel**

## Affichage à la première personne

Une

**Essentielle**

## Affichage d’une scène

Une scène sera affichée où chaque joueur sera montré avec l’instrument joué près de lui (comme s’il jouait d’un vrai instrument). Un décor pourra être affiché en arrière.

**Optionnel**

## Nombre de joueurs

Le nombre de joueur maximal sera de 2.

**Essentiel**

## Deux pianos en simultané

Le nombre de joueur maximal sera de 2.

**Optionnel**

# Exigences non-fonctionnelles

[Pour cette section, vous pouvez enlever les sous-sections que vous considérez non pertinentes.]

## Utilisabilité

[Cette sous-section devrait inclure toutes les exigences affectant l’utilisabilité. Par exemple:

* spécifier le temps de formation requis à un utilisateur normal et utilisateur spécialisé (power user) afin de devenir productif pour effectuer certaines opérations
* spécifier un temps d’exécution mesurable pour des tâches typiques]

### Temps de formation

Un utilisateur normal doit pouvoir utiliser le logiciel facilement, avec une période d’apprentissage maximum de quelques minutes.

### Temps de délai du son

Le son de chaque note pour chaque instrument devra être joué avec un délai maximal de quelques millisecondes après que l’action soit effectuée par l’utilisateur.

### Temps de délai de l’image

La scène sera affichée avec un délai maximal de quelques de dizaines de millisecondes après que l’action soit effectuée par l’utilisateur.

## Fiabilité

[Cette sous-section devrait inclure toutes les exigences de fiabilité. Par exemple:

* Disponibilité – spécifier le pourcentage de temps de disponibilité et d’heures d’utilisation.
* Temps moyen entre pannes (Mean Time Between Failures) – spécifié en heures, en jours, en mois ou en années.
* Temps moyen jusqu’à la réparation (Mean Time To Repair) – combien de temps le système peut-il être hors d’usage après une panne?]

### Temps moyen entre pannes

Le temps moyen entre panne devra être très grand, avec un maximum de x panne par x temps. (1 panne par 9 heures?)

### Temps moyen jusqu’à la réparation

Le système devra être facilement réparable et/ou calibrable, avec un temps moyen de réparation d’au maximum quelques minutes. (Temps de reboot max 2 minutes)

## Performance

[Cette sous-section devrait inclure toutes les exigences de performance. Par exemple:

* Temps de réponse à une transaction (moyenne et maximale)
* Utilisation de ressources (mémoire, disque, communication, etc.)]

### Utilisation de ressource

Le logiciel devra utiliser le moins de ressources possibles, considérant que les ordinateurs utilisés seront très probablement des ordinateurs portables à puissance limitée (spécialement au niveau de la carte graphique). De plus, le temps de traitement influera probablement sur les délais et doit donc être minimal.

On peut se permettre une utilisation excessive de CPU.

## Maintenabilité

[Cette sous-section devrait inclure toutes les exigences amélioration la maintenabilité du système, incluant les normes de codage, les conventions de nommage, l’accès de maintenance, etc.]

### Calibrage facile

Rayonnement N shits.

### Normes de codage

Google style guide.

## Contraintes de conception

[Cette sous-section doit indiquer toute contrainte de conception au système développé, comme le langage de programmation, le processus logiciel, l’achat de composantes, les libraires de classes, etc.]

### Langage de programmation

Le langage utilisé sera le C#, et le C++.

### Librairies utilisées

Le moteur 3D Unity sera utilisé ainsi que diverses librairies en lien avec le capteur et/ou avec la tranmission réseau.

OpenCV, etc

### Processus

Le processus utilisé sera une variation des processus agiles. Ce processus est défini plus en détails dans un schéma.

### Achat de composantes

Blah blah

## Exigences de la documentation usager en ligne et du système d’assistance

Barre d’instruments en haut pour savoir les instruments possibles à jouer.

# ANNEXE A: Glossaire

[Si vous n’utilisez pas cette annexe, veuillez la supprimer complètement.]

|  |  |
| --- | --- |
| **Terme** | **Description** |
| SDK | *Software Development Kit* |
| FPS | Vue à la première personne (*First person shooter*) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |