



PACT 2015-2016

Rapport d'avancement du groupe "3.1" "Fituring"



Date de mise à jour : 02/12/2015

Camille Demason, Khadija El Attar, Stanislas Hannebelle, Karim Khandid, Thibaud Lemaire, Alexandre Poinso, Robin Shin, Jamal Skouri

Tuteur : Michel Roux

Encadrant génie logiciel : Soumia Dermouche

Sommaire

1	RESUME DU SUJET CHOISI.....	5
2	ABSTRACT.....	5
3	ETUDE D'ANTERIORITE ET JUSTIFICATION DE LA PROPOSITION	6
3.1	DESCRIPTION DE LA PROPOSITION.....	6
3.2	DESCRIPTION DE L'ETAT DE L'ART	7
4	SCENARIO D'USAGE	9
5	ARCHITECTURE DU PROJET	11
5.1	SCHEMA D'ARCHITECTURE	11
5.2	DESCRIPTION DES BLOCS	12
5.2.1	KINECT.....	12
5.2.2	API KINECT SERVEUR – CLIENT ACQUISITION SQUELETTE	12
5.2.3	BASE DE DONNEES	12
5.2.4	DETECTION DES MOUVEMENTS (CLASSIFICATION) :.....	12
5.2.5	DETECTION DU RYTHME ET DU TEMPO	12
5.2.6	DETECTION DE RICHESSE DE LA DANSE	12
5.2.7	SEQUENCEUR MUSICAL	12
5.2.8	SYNTHESE AUDIO	12
5.2.9	APPLICATION ANDROID	12
5.2.10	SERVEUR	13
5.3	DESCRIPTION DES INTERFACES	13
5.3.1	INTERFACEBLOC UTILISATEUR – KINECT	13
5.3.2	INTERFACEBLOC KINECT - SERVEUR C & API KINECT	13
5.3.3	INTERFACEBLOC API KINECT & SERVEUR C – CLIENT JAVA.....	13
5.3.4	INTERFACEBLOC CLIENT JAVA – DETECTION DES MOUVEMENTS	13
5.3.5	INTERFACEBLOC CLIENT JAVA – DETECTION DU RYTHME.....	13
5.3.6	INTERFACEBLOC CLIENT JAVA – DETECTION RICHESSE DE LA DANSE.....	13
5.3.7	INTERFACEBLOC DETECTION DES MOUVEMENTS – BDD	13
5.3.8	INTERFACEBLOC DETECTION DES MOUVEMENTS – CHOIX DES SONS.....	14
5.3.9	INTERFACEBLOC DETECTION DES MOUVEMENTS – SEQUENCEUR MUSICAL	14
5.3.10	INTERFACEBLOC CHOIX DES SONS – BDD	14
5.3.11	INTERFACEBLOC SEQUENCEUR MUSICAL – SYNTHESE AUDIO	14

5.3.12	INTERFACEBLOC DETECTION DU RYTHME ET DU TEMPO – SEQUENCEUR MUSICAL/SYNTHESE AUDIO	14
5.3.13	INTERFACEBLOC DETECTION DE LA RICHESSE DE LA DANSE – SEQUENCEUR MUSICAL	14
5.3.14	INTERFACEBLOC FICHIER SONS – SYNTHESE AUDIO	15
5.3.15	INTERFACEBLOC SYNTHESE AUDIO – HAUT-PARLEURS	15
5.3.16	INTERFACEBLOC UTILISATEUR – APPLICATION ANDROÏD	15
5.3.17	INTERFACEBLOC APPLICATION ANDROÏD – SERVEUR	15
5.3.18	INTERFACEBLOC API KINECT – SERVEUR	15
5.3.19	INTERFACEBLOC SERVEUR – SEQUENCEUR MUSICAL	15
5.3.20	INTERFACEBLOC SERVEUR – SYNTHESE AUDIO	15
5.4	DIAGRAMME DE SEQUENCE	16
5.5	INTERFACE UTILISATEUR GRAPHIQUE	17
5.6	TABLEAU DETAILLE DES TACHES	18
6	ORGANISATION DU PROJET	19
6.1	DIAGRAMME DE PLANIFICATION TEMPOREL DES TACHES	19
6.2	REPARTITION DES ELEVES PAR MODULE	19
6.3	PLANS DE TEST	19
6.4	DIAGRAMME D’AVANCEMENT DES TACHES	19
7	BIBLIOGRAPHIE	21
ANNEXE A.	FICHE D’IDENTITE DU GROUPE	22
ANNEXE B.	MODIFICATIONS	23
B.1.	MODIFICATIONS DE FOND	23
B.2.	MODIFICATIONS DU RAPPORT	23
B.2.1.	MODIFICATIONS DU RAPPORT AU PAN2	23
B.2.2.	MODIFICATIONS DU RAPPORT AU PAN3	23
B.2.3.	MODIFICATIONS DU RAPPORT AU PAN4	23
ANNEXE C.	COMPTES RENDUS DE REUNIONS	24
C.1.	CR DU 5/10/2015	24
C.2.	CR DU 29/09/2015	25
	MEMBRES DU GROUPE	25
	DECOUVERTE DES TROIS THEMES PACT DE CETTE ANNEE	25

PREMIERES PROPOSITIONS DE SUJETS	25
ROBIN :	26
ALEXANDRE :	26
THIBAUD :	26
STANISLAS :	26
KHADIJA :	26
CAMILLE :	26
ORGANISATION DES SUJETS PROPOSES	26
REPARTITION DE TECHNOLOGIES CONNUES	27
<u>ANNEXE D. MODULE KINECT</u>	<u>30</u>
<u>ANNEXE E. MODULE CLASSIFICATION</u>	<u>32</u>
<u>ANNEXE F. DETECTION DU RYTHME</u>	<u>33</u>
<u>ANNEXE G. MODULE SYNTHESE AUDIO.....</u>	<u>34</u>
<u>ANNEXE H. MODULE ANDROÏD</u>	<u>35</u>
<u>ANNEXE I. MODULE SES.....</u>	<u>37</u>
<u>ANNEXE J. MODULE INTEGRATION ET TESTS.....</u>	<u>39</u>

1 Résumé du sujet choisi

Lorsqu'une personne danse, elle est habituée à bouger en fonction du rythme de la musique, selon son style et l'humeur qu'elle dégage. On a là un procédé où un certain son va provoquer une façon de danser spécifique. Pour notre projet PACT, nous voulons essayer d'inverser ce processus : le danseur peut laisser s'exprimer son corps sans contrainte, et notre appareil, appelé Fituring, sera capable de générer une musique originale qui saura s'adapter à chaque mouvement de l'utilisateur et à son style. Fituring aura la forme d'un boîtier contenant une kinect et un petit ordinateur capable de traiter les données de mouvement en sortie de la kinect et de les convertir en un son adéquat. Le but de ce projet est d'offrir à l'utilisateur un nouveau mode de création artistique passant par le corps, avec la possibilité d'un effet d'apprentissage de manière qu'il peut prévoir et moduler la musique selon son envie, et non pas aléatoirement. Nous vivons dans une société où la communauté et les réseaux sociaux ont une grande importance : Fituring ne laissera pas ces domaines de côté car il sera accompagné d'une application où l'utilisateur pourra retrouver ses anciennes créations et les partager sur son réseau social préféré.

2 Abstract

Fituring is a whole new device that will revolutionize the way you dance. People are used to adapt their moves to the music they are listening to. Now, they can control the music, its rhythm and its style according to their moves : your body is a new extremely complex instrument. Your legs will command the tempo, your arms and the top of your body will be in charge of the sound. For more and more requirements, you can chose a few options on the smartphone application, like a precise style of music or the volume. When you will get tired of dancing this much, the application will allow you to live those incredible moments again : Fituring has captured your moves and registered the music created so that you can keep videos of your performance and share it with your friend on social networks.

3 Étude d'antériorité et justification de la proposition

3.1 Description de la proposition

Lors de nos premières recherches de sujet pour le projet PACT, une idée s'est démarquée des autres, rentrant dans le thème : « conversion de gestes en son ». Il s'agissait de la réalisation d'un instrument de musique corporel et donc sans objet à manipuler. Le principe de ce projet était d'assigner chaque mouvement possible du corps à une certaine note ou un certain rythme. Par exemple, la position des bras aurait contrôlé la hauteur de la note et la vitesse des mouvements aurait défini le rythme. Apprendre à jouer à notre instrument de musique revenait alors à maîtriser des mouvements, ce qui aurait permis de jouer n'importe quelle partition de musique. Toutefois, nous nous sommes rendu compte que ce type de fonctionnement nécessiterait un apprentissage difficile de la part de l'utilisateur, de la même manière que pour un instrument déjà existant. Ce point aurait rendu l'utilisation de notre appareil difficile, entraînant donc probablement une faible demande de la part des consommateurs : la réalisation de notre instrument n'était donc pas justifiée. Il nous a donc fallu repenser le fonctionnement de l'appareil pour le rendre plus intuitif : d'où l'idée d'inventer plutôt un dispositif demandant simplement à l'utilisateur de danser librement ; la création de la musique correspondante sera entièrement gérée par l'appareil.

Cette nouvelle idée nous convenant à tous, répondant parfaitement au thème PACT et promettant un résultat qui plaira à un large public, nous avons alors commencé à définir les moyens dont nous aurons besoin pour réaliser un projet si complexe. Il nous est tout de suite paru évident que l'acquisition de l'image et des mouvements de l'utilisateur se ferait grâce à une kinect qui fournirait en sortie ces données sous forme de squelette. La première difficulté est alors apparue au sujet du traitement de ce squelette qui n'est pas faisable par le biais d'un programme en java : nous avons donc déterminé le besoin d'un module pour nous apprendre à utiliser la kinect et convertir ses données grâce à un programme en langage C pour pouvoir les traiter par la suite.

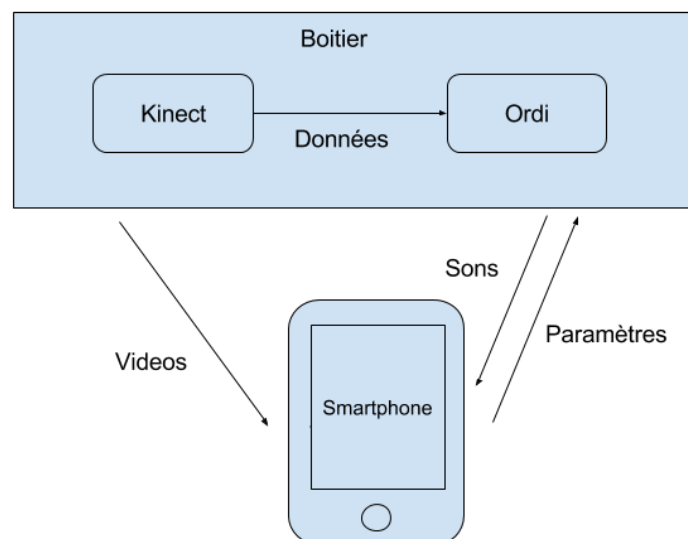
Par la suite, la difficulté majeure que nous avons mise en évidence est la façon de convertir des gestes en une musique qui conviendrait : sur quel critères pouvons-nous associer un mouvement à une caractéristique musicale telle que la hauteur des sons ou le rythme ? Comment choisir les sons adéquats dans une grande bibliothèque ? Comment reconnaître des mouvements précis ? Et comment générer une musique qui plaira ?

En élaborant le scénario d'usage, nous nous sommes rendu compte que notre système serait intéressant à utiliser en soirée, surtout pour relancer la danse quand personne ne semble vouloir se proposer. Cependant la kinect ne permet de détecter qu'une à quatre personnes. Sans oublier qu'en classant un ensemble de gestes prédéfinis, ils seront effectués de manières différentes par les consommateurs ce qui ne nous facilitera pas la tâche d'acquisition et pourrait même engendrer une musique qui ne convient pas au style de danse de l'utilisateur.

Ce qui nous semble le plus dur à réaliser est le fait de gérer les transitions au cas où la personne décide soudainement de changer son style, richesse, ou vitesse de danse. Il nous faudrait réaliser une continuité dans la musique quel que soit le changement.

Pour que notre appareil réalise toutes les fonctionnalités voulues et puisse nous permettre de trouver des solutions adéquates à nos problèmes, nous allons concevoir notre dispositif comme un boîtier composé d'une caméra kinect et d'un mini-ordinateur contrôlés par un smartphone Android. L'utilisateur choisit ses paramètres à travers l'application Android qui les transmet à l'ordinateur, la kinect permet d'analyser la danse de l'utilisateur. Le système génère automatiquement la musique qui correspond à ses mouvements. Il diffuse la musique ainsi générée, et permet de l'enregistrer pour la diffuser plus tard. Enfin, des fonctionnalités sociales seront incluse pour pouvoir partager les musiques créées avec la vidéo du danseur correspondante.

Ci-dessous un schéma montrant l'architecture de notre système :



3.2 Description de l'état de l'art

Notre projet PACT se propose de développer un instrument qui crée une piste musicale à partir de la danse de l'utilisateur.

L'idée de créer de la musique à partir de mouvement et sans contact avec l'instrument, bien qu'originale, n'est pas nouvelle. En effet, en 1919, le russe Léon Thérémine invente l'un des tout premiers instruments de musique électronique : le thérémine (cf http://www.sonelec-musique.com/electronique_realisations_th...). Cet instrument est composé de deux antennes.

Chacune d'elle permet de relever la position d'une main. De façon générale, la hauteur de la main gauche donne le volume tandis que celle de la main droite donne la note à jouer par l'instrument. Cet instrument est révolutionnaire pour l'époque cependant, il ne permet pas de jouer une piste musicale riche mais uniquement un son également, à l'inverse de notre projet, il se base sur des mouvements précis que réalise l'utilisateur et non sur sa danse.

L'idée d'un instrument de musique virtuel a également été développée. Par exemple, en utilisant la technologie Kinect pour capturer des mouvements, des étudiants de Georgia Tech ont créé un instrument de musique virtuel à dessiner (https://www.youtube.com/watch?v=CEd4dS_pOVM). L'utilisateur doit dessiner des formes sur une feuille de papier, ensuite lorsqu'il touchera l'une de ses formes un son particulier sera joué. Il peut alors composer en touchant les différentes notes. Cependant, là aussi ce n'est pas une danse qui crée la musique mais uniquement des gestes.

Le compositeur Chris Vik a développé un système qui capte les mouvements d'un danseur et qui crée de la musique à partir de cette danse (<https://www.youtube.com/watch?v=qXnLxi2nZrY>). Dans ce système c'est également la technologie Kinect qui permet la capture des mouvements du danseur. Cependant, l'utilisateur de cet instrument est un danseur professionnel. A l'inverse, notre projet se veut accessible à tous et notre système devra être capable de produire une piste musicale de qualité même si les pas de danse de l'utilisateur ne sont pas dignes d'un grand danseur. Enfin, le système Kinect Bomba, créé par Stephanie Gil et Sam Prentice permet la création d'une piste de musique à partir des mouvements de l'utilisateur (<http://people.csail.mit.edu/prentice/bomba/>). Il est même capable de prendre en compte les mouvements de deux danseurs en même temps. Ce système se rapproche de notre projet puisque son utilisation est assez intuitive pour l'utilisateur. Cependant, il ne permet uniquement la production d'un style musical latin alors que dans notre projet nous souhaitons que l'utilisateur puisse choisir lui-même le style musical à jouer par l'utilisateur. Ainsi, il existe déjà des systèmes qui créent des sons à partir de mouvement cependant notre projet souhaite apporter plus de liberté à l'utilisateur par rapport au choix de style musical ou par rapport aux gestes qu'il peut réaliser.

4 Scénario d’usage

17h00 : Paola, étudiante, rentre de cours après une dure journée d’examens. Elle a besoin d’un moment de détente et active sa playlist. Sa musique préférée se lance alors : “Happy” de Pharrell Williams. Immédiatement, une poussée d’adrénaline l’enivre et lui donne envie de danser. Après 4’07’’ de pur plaisir, la transition est brutale : la chanson suivante est “Quelqu’un m’a dit” de Carla Bruni Sarkozy. Paola est alors coupée dans son élan ; elle l’avait déjà bien trop écoutée cette musique. Et puis l’extrême douceur et lenteur de la chanson ne correspondait plus du tout à son humeur. Impossible de continuer à danser. La déprime des examens remonte à la surface, rien ne va plus. Paola est lassée et en a assez de ces playlists où les mêmes chansons tournent continuellement en boucle et ne veut plus dépendre du rythme de la musique : elle voudrait renverser le processus afin que ce soit la musique qui s’adapte à son humeur, à son rythme.

21h00 : Paola se rend chez son ami Paolo qui veut festoyer. En arrivant sur les lieux, elle est déçue, la fête ne semble pas encore commencée : aucune musique pour danser car par timidité, personne n’a osé commencer à danser. Paola, de plus en plus frustrée par sa journée, s’étonne auprès de son ami : “c’est quoi cette soirée qui bouge pas?” Paolo lui lance alors son plus grand sourire et lui rétorque : “Je fais une expérience. A toi de mettre l’ambiance ! Danse et tu verras bien ce qui se passera”. Mise au défi, elle esquisse un mouvement timide d’épaule qui entraîne un beat sonore. Surprise, elle tente cette fois un enchaînement de pas qui lui plaît. Immédiatement, un son de batterie parfaitement en rythme et synchronisé avec ses pieds retentit. Progressivement, c’est tout le corps de Paola qui se met à se mouvoir, et déferle une véritable symphonie électrique : tel un maestro son corps prend les rênes de toute la musique qui envahit la salle. Étonnés par sa performance, ses amis la rejoignent et se câlent à son rythme et progressivement, la musique s’enrichit. Une fois l’ambiance installée et la soirée démarrée, la musique est redirigée vers une playlist de musiques connues.

Sans le savoir, Paolo a réalisé le souhait émis par son amie un peu plus tôt dans la journée : il lui explique donc que tout cela est possible grâce au nouvel appareil *Fituring*.

Dès le lendemain, Paola s’empresse d’acquérir ce petit bijou de technologie et de l’essayer chez elle. Elle découvre donc plus en détail *Fituring*, un séquenceur musical autonome et hautement interactif. Il crée et diffuse de la musique qui s’adapte à la volée aux pas de danse de ses utilisateurs. Le dispositif est muni d’une caméra Kinect qui analyse les mouvements, relié à un ordinateur qui permet de traiter l’information des mouvements en son. Ceux-ci sont discriminés par un algorithme de classification liée à une base de donnée. Le système pioche ensuite dans une large bibliothèque de sons élémentaires pour créer une musique harmonieuse grâce à un procédé de boucle (“loop”) et parfaitement adaptée aux pas des danseurs.

Pour choisir les sons les plus adaptés, *Fituring* analyse plusieurs caractéristiques de la danse tels que le rythme, l’amplitude et la richesse des mouvements. Il y a une première phase d’initialisation du rythme pendant laquelle l’utilisateur doit uniquement bouger ses jambes afin de calculer une moyenne de tempo, qui peut évoluer si le danseur change de rythme. Ainsi, la musique produite est parfaitement synchronisée aux mouvements de l’utilisateur et

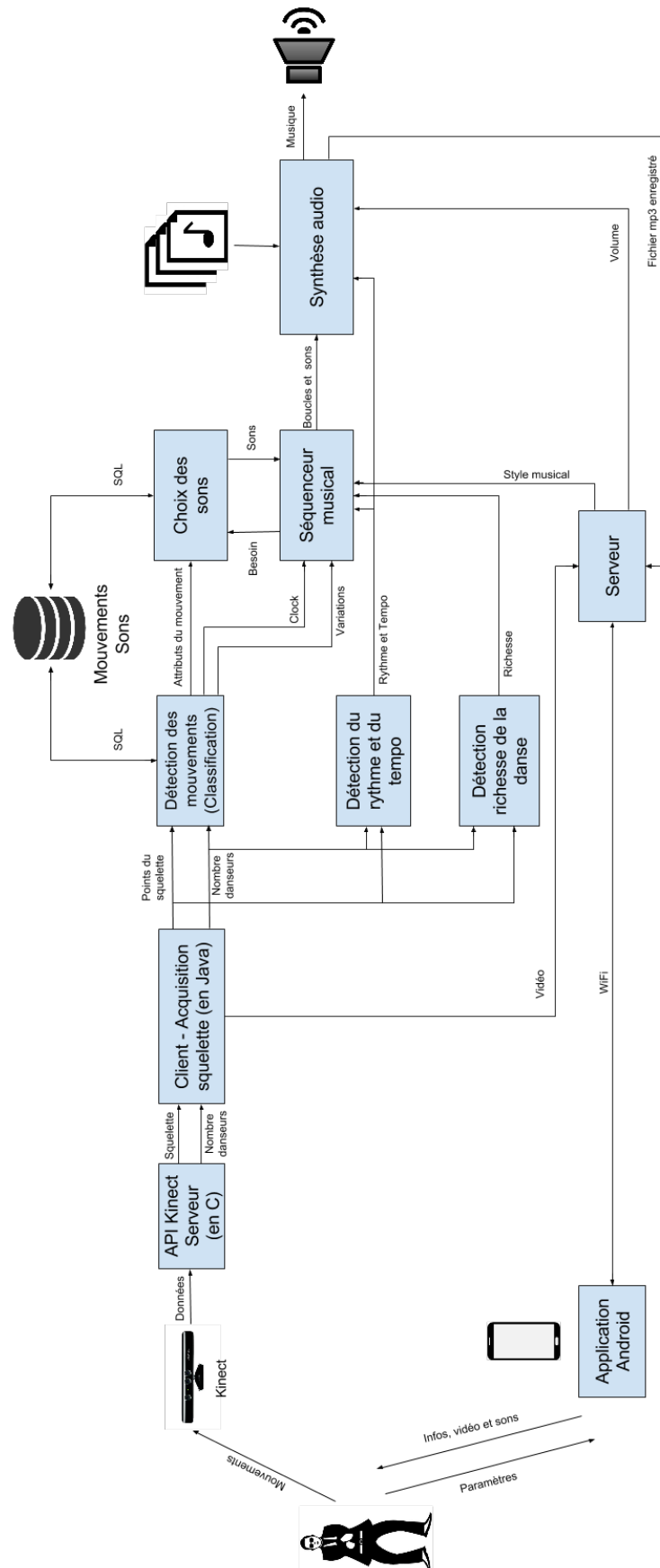
elle s'enrichit sur la base de structures musicales prédéfinie au fur et mesure qu'il ajoute des mouvements.

Paola prend également connaissance de l'application smartphone liée à l'outil, qui lui apprend via un tutoriel interactif comment utiliser *Fituring* au mieux. Cette application permet également de régler différentes options pour des types de musique différents, elle peut enregistrer les musiques créées.

Paola se découvre alors un réel talent de création artistique lié à la musique et à la danse, elle utilise *Fituring* lorsqu'elle a envie de danser, lorsqu'elle veut découvrir une nouvelle musique originale et même lorsqu'elle fait du sport dans son salon. *Fituring* devient alors le meilleur allié du fitness.

5 Architecture du projet

5.1 Schéma d'architecture



5.2 Description des blocs

5.2.1 Kinect

La Kinect fournit des informations de profondeur et les images issues de sa caméra RGB, en temps réel (environ 30Hz). Les données transitent via un bus USB.

5.2.2 API Kinect Serveur - Client Acquisition squelette

Acquisition et traitement des données en C, grâce à l'**API Kinect de Microsoft**. Transfert de ces données via une architecture Client/Serveur à notre programme Java. Pour cela, coté java, nous utiliserons la bibliothèque **java.net**.

5.2.3 Base de données

Cette base de données contient deux tables : la première contient les mouvements usuels acquis à travers le module SES observation et caractérisés par leur style, les membres mis en mouvements, etc... La deuxième table contient les sons élémentaires disponibles, c'est-à-dire les briques de base qui serviront à construire la musique. Nous utiliserons la bibliothèque **java.sql** pour accéder à cette BDD.

5.2.4 Détection des mouvements (classification) :

La fonction du module Classification est d'identifier les mouvements effectués par l'utilisateur en se basant sur les mouvements préenregistrés avec l'approche probabiliste nécessaire puisque tous les utilisateurs ne font pas le même mouvement de la même manière et avec la même précision.

5.2.5 Détection du rythme et du tempo

La détection du tempo se fera à partir des mouvements des pieds et celle du rythme à partir des mains.

5.2.6 Détection de richesse de la danse

Ce bloc évalue la variation de position des membres du danseur. Cette « richesse » de la danse sera représentée par un curseur, plus la variation est élevée plus la musique contient des sons élémentaires différents.

5.2.7 Séquenceur musical

Le séquenceur musical détermine les sons à placer dans la musique générée à partir des informations sur la dans qui lui parviennent, des sons déjà utilisés dans la musique, et des exigences de l'utilisateur. Il exprime un besoin au bloc de choix des sons (par exemple « il me faut une ligne de basses profonde et électronique »).

5.2.8 Synthèse audio

Ce bloc construit la musique à partir des boucles et sons choisis par le séquenceur. Il réalise la lecture, la diffusion et l'enregistrement du son. Nous utiliserons la bibliothèque **java.sound** pour la manipulation des sons et **java.io** pour la manipulation de fichiers.

5.2.9 Application Android

L'application Android permet à l'utilisateur de communiquer le volume et style musical souhaités, et de parcourir les anciennes musiques précédemment.

5.2.10 Serveur

Le serveur sert de passerelle entre l'application mobile et le système. Il met à disposition les sons et les images sauvegardées et permet d'effectuer des réglages du son. Nous utiliserons la bibliothèque **java.net** pour la gestion des sockets.

5.3 Description des interfaces

5.3.1 InterfaceBloc Utilisateur - Kinect

La Kinect filme et mesure les mouvements de l'utilisateur grâce à une caméra RGB classique et deux capteurs de profondeur IR.

5.3.2 InterfaceBloc Kinect - Serveur C & API Kinect

Ces deux blocs communiquent via un bus USB. Par ce support, la Kinect transmet à son API des données video, audio et des informations de profondeur à une fréquence d'environ 20Hz. Ces données sont en l'état difficile à utiliser, d'où l'intérêt d'utiliser l'API Kinect pour en extraire l'information.

5.3.3 InterfaceBloc API Kinect & Serveur C - Client Java

Il s'agit là de faire le lien entre l'API Kinect (codée en C) et notre programme Java. On récupère un ensemble de 20 points (par danseur) formant le squelette de l'utilisateur, toujours à une fréquence d'environ 20Hz. On récupère également le nombre de danseurs actuellement détectés par la Kinect. Les données sont échangées suivant une architecture Client-Serveur, au sein d'un même ordinateur : le processus codé en C faisant office de serveur auquel se connecte le client Java.

5.3.4 InterfaceBloc Client Java - Détection des mouvements

Ces blocs échange un objet « squelette » contenant la dernière localisation spatiale de l'ensemble des articulations du squelette de(s) l'utilisateur(s) dans le référentiel de la Kinect. Cet objet contient également le nombre de danseurs actuellement détectés.

5.3.5 InterfaceBloc Client Java - Détection du rythme

Ces blocs échange un objet « squelette » contenant la dernière localisation spatiale de l'ensemble des articulations du squelette de(s) l'utilisateur(s) dans le référentiel de la Kinect. Cet objet contient également le nombre de danseurs actuellement détectés.

5.3.6 InterfaceBloc Client Java - Détection richesse de la danse

Ces blocs échange un objet « squelette » contenant la dernière localisation spatiale de l'ensemble des articulations du squelette de(s) l'utilisateur(s) dans le référentiel de la Kinect. Cet objet contient également le nombre de danseurs actuellement détectés.

5.3.7 InterfaceBloc Détection des mouvements - BDD

Le bloc « Détection des mouvements » interroge la table « mouvements » de la base de données afin d'identifier le mouvement actuellement réalisé par l'utilisateur. L'interrogation de la base de données d'effectue par requêtes SQL, en réponse, le serveur de BDD renvoie l'ensemble des mouvements probable ainsi que leurs attributs (quelle partie du corps, amplitude du mouvement, saccades, type de danse, etc...)

5.3.8 InterfaceBloc Détection des mouvements - Choix des sons

Ces blocs s'échangent des informations sur les attributs des mouvements détectés. Ces attributs prennent la forme de « curseurs » (un pourcentage) évaluant, par exemple, l'utilisation du bras gauche, la fluidité de la danse, si le mouvement est plutôt vertical, etc... Le bloc « Detection des mouvements » en détermine les valeurs à partir des attributs préenregistrés dans la BDD mouvement, pondérés par la probabilité de détection dudit mouvement. Tous ces curseurs sont stockés dans un objet « mouvements » mis à jour à une fréquence proche de 20Hz et récupéré par le bloc « Choix des sons ».

5.3.9 InterfaceBloc Détection des mouvements - Séquenceur musical

Le bloc « Détection des mouvements » fournit deux types d'informations au séquenceur musical. D'une part il permet de synchroniser la danse et le son en indiquant les « temps forts de la danse » (typiquement un saut, particulièrement dans le cas d'une danse saccadée). Cette synchronisation se fait par l'appel d'une méthode du bloc « Séquenceur musical ».

En plus de la synchronisation, le bloc « Détection des mouvements » fournit une estimation des changements de mouvement, permettant au séquenceur de passer à une autre phase musicale.

5.3.10 InterfaceBloc Choix des sons - BDD

Le bloc « Choix des sons » interroge la table « sons » de la base de données afin de filtrer les sons correspondant à la danse et à la demande du séquenceur musical. L'interrogation de la base de données d'effectue par requêtes SQL, en réponse, le serveur de BDD renvoie l'ensemble des sons utilisables, leurs caractéristiques et leur chemin d'accès.

5.3.11 InterfaceBloc Séquenceur musical - Synthèse audio

Le bloc « Séquenceur musical » fournit le chemin d'accès des sons à jouer ainsi que les informations permettant le positionnement temporel (dans des boucles) de ces sons. Concrètement, chaque boucle prend la forme d'un objet, avec comme attributs le chemin du son, la fréquence de ceux-ci au sein d'une mesure, etc...

5.3.12 InterfaceBloc Détection du rythme et du tempo - Séquenceur musical/Synthèse audio

Le bloc « Détection du rythme et du tempo » fournit aux deux blocs audio les informations détectées dans la danse. Concrètement, un objet « tempo » est créé et est mis à jour en temps réel. En plus du rythme et du tempo, il contient une indication sur leurs variations afin de détecter des changements de phases musicales.

5.3.13 InterfaceBloc Détection de la richesse de la danse - Séquenceur musical

Le bloc « Détection de la richesse de la danse » fournit une estimation de la richesse de la danse sous la forme d'un « curseur » (entre 0 et 100) encapsulé dans un objet « richesse ». Cette information permet au séquenceur musical de choisir combien de sons jouer en même temps.

5.3.14 InterfaceBloc Fichier sons - Synthèse audio

Le bloc « Synthèse audio » récupère les fichiers sons élémentaires à jouer à partir de leur chemin d'accès récupéré dans la BDD.

5.3.15 InterfaceBloc Synthèse audio - Haut-parleurs

Le bloc « Synthèse audio » utilise la carte son du PC pour restituer en direct la musique générée sur des haut-parleurs.

5.3.16 InterfaceBloc Utilisateur - Application Androïd

Via l'application Androïd, l'utilisateur peut régler à l'avance le style musical qu'il souhaite, le volume, etc... Il peut également récupérer et jouer une musique qu'il a préalablement enregistré grâce à Fituring.

5.3.17 InterfaceBloc Application Androïd - Serveur

L'application Androïd communique avec le serveur via le protocole IP sur une liaison WiFi AdHoc. Les échanges sont bidirectionnels.

5.3.18 InterfaceBloc API Kinect - Serveur

Dès le début de la danse, le serveur enregistre l'image de la caméra RGB de la Kinect afin de pouvoir la superposer à la musique créée lors de la restitution.

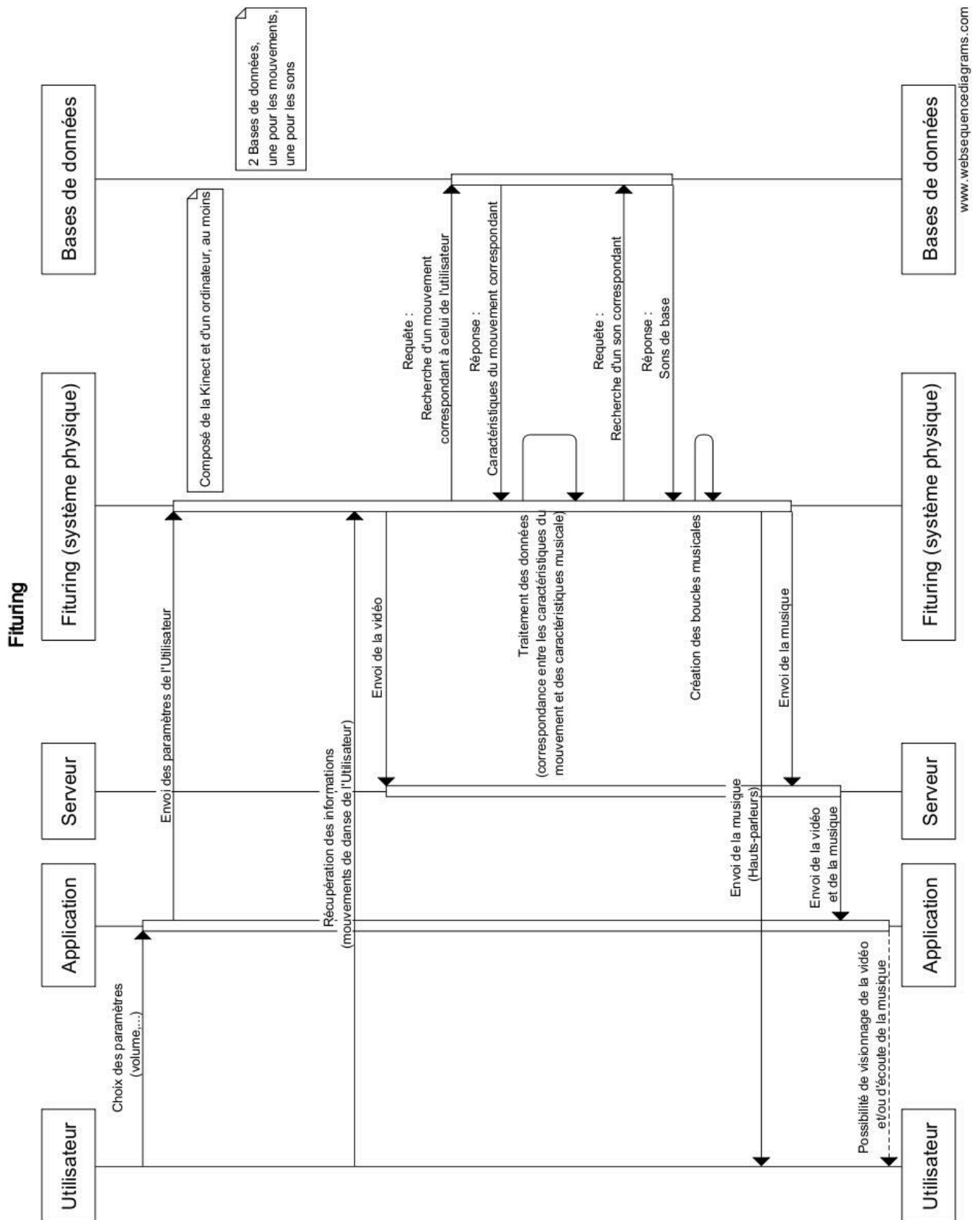
5.3.19 InterfaceBloc Serveur - Séquenceur musical

Conformément aux souhaits de l'utilisateur, le serveur fournit au séquenceur musical le style de musique à générer. Cela se fait par l'appel d'une méthode de l'objet « musique ».

5.3.20 InterfaceBloc Serveur - Synthèse audio

Conformément aux souhaits de l'utilisateur, le serveur règle le volume de la restitution audio par l'appel d'une méthode. A la fin de la musique, le bloc « synthèse audio » fournit au serveur le fichier audio généré pour pouvoir le réécouter.

5.4 Diagramme de séquence



5.5 Interface utilisateur graphique



5.6 Tableau détaillé des tâches

Tâche	Description	Démontrée au PAN3	Intégrée au PAN4
T1	Synthèse Audio (Jamal.Skouri , Stan Hannebelle , Khadija El Attar)		
T1.1	Programme qui lit un fichier son	X	X
T1.2	Programme qui joue un fichier son en boucle et en rythme	X	X
T1.3	Programme qui enregistre la musique dans un fichier	X	
T1.4	Comparaison des caractéristiques		
T2	Détection de rythme (Stan Hannebelle, Khadija El Attar)		
T2.1	Découpage signal sonore en bande fréquentielle	X	X
T2.2	Détection du tempo	X	
T2.3	Changement de tempo		
T3	Kinect Squelette (Thibaud Lemaire, Robin Shin)		
T3.1	Application test de suivi de l'ensemble du corps	X	X
T3.2	Définition du format d'échange des données entre C et Java	X	X
T3.3	Application client-serveur entre Java et C pour l'échange de données via sockets TCP	X	X
T3.4	Suivi du corps complet C → Java intégré dans le prototype allégé	X	
T3.5	Caractérisation de l'utilisation de la Kinect pour notre projet (Bilan)	X	
T4	Androïd (Camille Demasson , Khadija El Attar)		
T4.1	Réaliser une application spécifique (à définir en fonction des besoins du projet)	X	X
T4.2	la communication via le réseau, par exemple selon le protocole HTTP.	X	X
T4.3	Décrire le fonctionnement et l'intégration du module dans le prototype allégé	X	X
T4.4	Analyser comment le module est intégré dans le prototype	X	
T5	Intégration & test (Alexandre Poinso, Thibault Lemaire, Camille Demasson)		
T5.1	Rapport sur l'état de l'intégration des modules	X	X
T5.2	Résultat des tests	X	X

Tableau 1 - Description des tâches

6 Organisation du projet

6.1 Diagramme de planification temporel des tâches

Cf. annexe

6.2 Répartition des élèves par module

Nom Module	Module Kinect	Module Classification	Module Détection du rythme	Module Synthèse audio	Module Androïd	Module SES	Test & Intégration
Nom Expert	Jean Lefeuvre	Slim Essid	Bertrand David	Jean-Claud Dufourd	Jean- Claude Dufourd	Julien Morel	Soumia Dermouche
Eleve1	Thibaud Lemaire	Thibaud Lemaire	Khadija El Attar	Jamal Skouri	Camille Demasson	Jamal Skouri	Thibaud Lemaire
Eleve2	Robin Shin	Karim Khandid	Stan Hannebelle	Stan Hannebelle	Khadija El Attar	Karim Khandid	Alexandre Poinsot
Eleve3		Robin Shin		Khadija El Attar		Stan Hannebe lle	Camille Demasson

Tableau 2 - Répartition élèves-modules

6.3 Plans de test

Note : à ne compléter qu'à partir de PAN2.

Vous allez travailler sur chaque bloc de votre projet, et qu'il soit informatique, électronique ou matériel, vous allez devoir faire du test :

- tester que le bloc que vous venez de finir fait ce qu'il faut ;
- tester que le bloc fonctionne avec les blocs en amont ou en aval dans l'architecture ;
- tester que les performances sont acceptables...
- et plus globalement, tester que le projet « marche ».

Vous allez devoir faire ce travail sur le prototype allégé, puis sur le prototype final. C'est un travail dans le module « intégration et tests ».

Cette section rassemble les plans de test du proto allégé et du proto final. C'est une liste des tests à effectuer, sous la forme, pour chaque test :

- situation/contexte
- action ou entrée à appliquer
- réaction ou sortie attendue.

6.4 Diagramme d'avancement des tâches

Note : à compléter à partir de PAN2

Mettre à jour le Tableau 3 à chaque PAN – Vous devez y résumer l’avancement de vos modules. L’avancement est mesuré par rapport à l’ensemble du travail à effectuer. L’avancement prévu correspond à l’avancement prévu au PANX, tel que renseigné dans votre diagramme temporel. L’avancement réel est ce que vous avez réellement atteint. Vous préciserez également le temps passé sur le module, afin d’apprendre à évaluer la durée des tâches. La description du travail effectué est brève mais vous devez absolument rencontrer votre expert (encadrant de module) pour qu’il détaille ses attentes en termes de rendus dans la fiche module qu’il vous remettra. Avant chaque PAN vous devrez le rencontrer pour récupérer une appréciation de votre travail. Lorsque vous estimerez votre module terminé, vous demanderez une soutenance afin d’être évalué et d’obtenir une note de module. La soutenance s’effectue devant au moins deux enseignants.

Module	Avancement Prévu (%)	Avancement Réel (%)	Temps passé (h)	Description brève du travail effectué, analyse des écarts constatés
MonModule	20%	20%	8	Les tâches x et y sont terminées – La tâche z est en préparation – Un rapport a été transmis à l’expert – N réunions ont eu lieu avec pour résultat, etc.

Tableau 3 - Avancement des modules

7 Bibliographie

Note : Liste des références exploitées. Une référence complète donne titre, auteur(s), date, journal, revue, source de publication, titre de conférence, numéro, pages. Une webographie est aussi envisageable : titre, auteur, date, page web

- [1] Aguirre, E. ; Jacqmot, C. ; Milgrom, E. ; Raucent, B. ; Soucisse, A. ; Trullemans, Ch. ; Vander Borcht, C. (2001) Devenir ingénieur par apprentissage actif, pédagogie par le projet, QPES, Brest.
- [2] Prigent, R. ; Bernard, H. ; Kozanitis, A. (2010). « Enseigner à l'université dans une approche-programme : guide à l'intention des nouveaux professeurs et chargés de cours. Canada : Presses Internationales Polytechnique »
- [3] Vitrine des projets PACT, <https://pact.wp.mines-telecom.fr/vitrine/>

Annexe A. Fiche d'identité du groupe

Consigne de rédaction : 1 page max rédiger collectivement un texte permettant de définir votre groupe et les personnalités qui le composent. Un bref descriptif individuel peut suivre (optionnel) la description collective.

Annexe B. Modifications

A compléter à partir de PAN2

B.1. Modifications de fond

Tableau des modifications de fond apportées au projet avec validation des experts et encadrant informatique

libellé / date	Description brève	Validé par :

B.2. Modifications du rapport

Vous noterez dans cette section les modifications apportées au rapport depuis le PAN précédent. Si votre planification temporelle a été modifiée, vous laisserez l'ancienne planification dans cette annexe.

B.2.1. Modifications du rapport au PAN2

B.2.2. Modifications du rapport au PAN3

B.2.3. Modifications du rapport au PAN4

Annexe C. Comptes Rendus de réunions

C.1. CR du 5/10/2015

Stan : Projet langue des signes

Bonne idée car utile et très ambitieux : beaucoup de mouvements, différences subtiles

Peut-être qu'on n'a pas le niveau... Coté synthèse de la voix intéressant. On peut dépasser facilement les 8 modules.

Grosse difficulté du sujet : détection des mouvements → Comment ? A distance ou avec un équipement ?

Grosse partie de programmation : traitement des données, synthèse de la voix, etc...

Question de Camille : la difficulté ne vient-elle pas plutôt de la connaissance du langage des signes ? Personne ne connaît cette langue parmi nous.

Khadija explique comment elle envisage la détection des mouvements par « pixellisation »

Robin pense que c'est impossible à faire à notre niveau

Thibaud : Le jeu des signes/musique de Camille

Ludique est très modulaire, on peut à peu près tout imaginer du moment que c'est un jeu

Robin : Skate streetview

Skate statique qui se déplace sur streetview grâce à l'inclinaison du skate
projet PACT déjà réalisé avec un vélo

Problème de la fluidité des images

Thibaud trouve ça trop facile. Camille trouve que la finalité et l'utilité du projet n'est pas super.

Jamal : Invention d'un instrument qu'on n'a pas besoin de toucher

C'est intéressant car d'habitude musique → danse alors que là c'est inversé

C'est difficile car il faut des gens qui maîtrisent la danse et la musique : il faut que ce soit cohérent entre danse et musique

Peut être réalisable avec des lasers qui sortent du sol

Stan pense qu'on peut mettre en place des palettes pour choisir un style musical afin de limiter le nombre et la précision des mouvements

Robin n'est pas emballé

Karim : Pas d'idée à défendre

Alexandre : Langage des signes

Camille : Lunettes GPS

Afficher le trajet sur les lunettes, ambitieux

Relier les lunettes à un Smartphone pour permettre de calculer le trajet

Difficulté : positionnement du trajet sur les lunettes

Khadija : Maillot de sport qui capte les mouvements

Mesure de l'intensité de l'effort du sportif

Capture de geste sportif précis pour les optimiser (proche des technologies de film 3D)

Existe déjà

Amélioration possible : augmenter le nombre de puce (pour quoi faire ?)

Difficulté du minimiser la taille de l'outil

Jamal : Chargeur de batterie par l'effort physique

Utile mais pas trop de rapport avec PACT

3 Projets Retenus :

- Langue des signes
- Corps instrument / Jeu mouvement-son
- Stylo réseau social permettant d'envoyer des dessins aux autres utilisateurs
-

C.2. CR du 29/09/2015

Membres du groupe

Le groupe 3.1 est composé des étudiants suivants : Jamal SKOURI, Robin SHIN, Alexandre POINSO, Thibaud LEMAIRE, Karim KHANDID, Stanislas HANNEBELLE, Khadija EL ATTAR SOFI et Camille DEMASSON. Seuls Karim et Jamal n'ont pas pu être présents.

Découverte des trois thèmes PACT de cette année

Nous avons la possibilité de réaliser notre projet PACT en respectant l'un des trois thèmes suivants :

- Du geste au son ;
- "L'invasion" virtuelle de la ville ;
- Le projet inédit.

Premières propositions de sujets

Chacun des membres présents de notre groupe a énoncé ses premières idées :

Robin :

- Skate connecté à street View permettant de se déplacer dans les rues virtuelles grâce à l'inclinaison et l'orientation du skate (1).

Alexandre :

- Voiture miniaturisée commandée par les gestes (2) ;
- Maillot sportif équipé de capteurs permettant de suivre les mouvements (rythme cardiaque, etc.) du sportif qui le porte (3).

Thibaud :

- Appareil (type mp3) qui adapte la musique écoutée selon son rythme par rapport à l'intensité d'un effort physique (4) ;
- Invention d'un instrument gyroscopique : le son produit dépend de l'orientation de l'instrument (5) ;
- Livraison de colis sur chariots qui s'orientent eux-mêmes grâce aux données cartographiques en ligne (6).

Stanislas :

- Tapis roulant entouré de la projection de décors (sur des écrans ou dans des lunettes) venant des données en ligne et permettant de s'orienter dans ces décors (7) ;
- Appareil qui règle à distance le style de musique à écouter selon les gestes effectués par l'utilisateur (8).

Khadija :

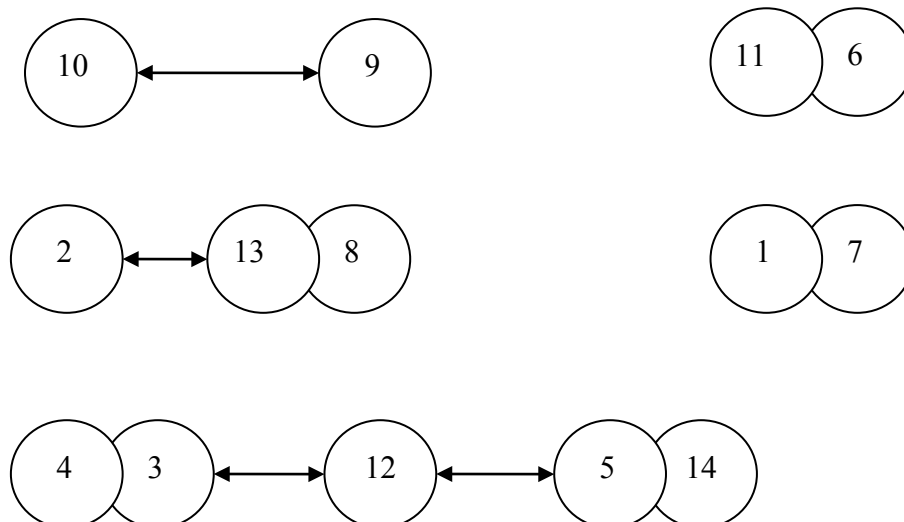
- Traduction de la langue des signes en paroles (idée de solution : capter et pixeliser le champ autour de la personne qui fait les gestes) (9) ;
- Traduction en paroles/son des besoins des plantes et des animaux en fonctions de leurs gestes (10) ;
- Lunettes GPS : lunettes portées pendant un trajet qui permet d'afficher l'itinéraire à suivre directement sur la route devant l'utilisateur (11).

Camille :

- Jeu : reproduire certains gestes particuliers pour jouer correctement une musique (12) ;
- Enceinte qui répond aux gestes (13) ;
- Invention d'un instrument que l'on n'a pas besoin de toucher : traduction d'une danse en musique (14).

Organisation des sujets proposés

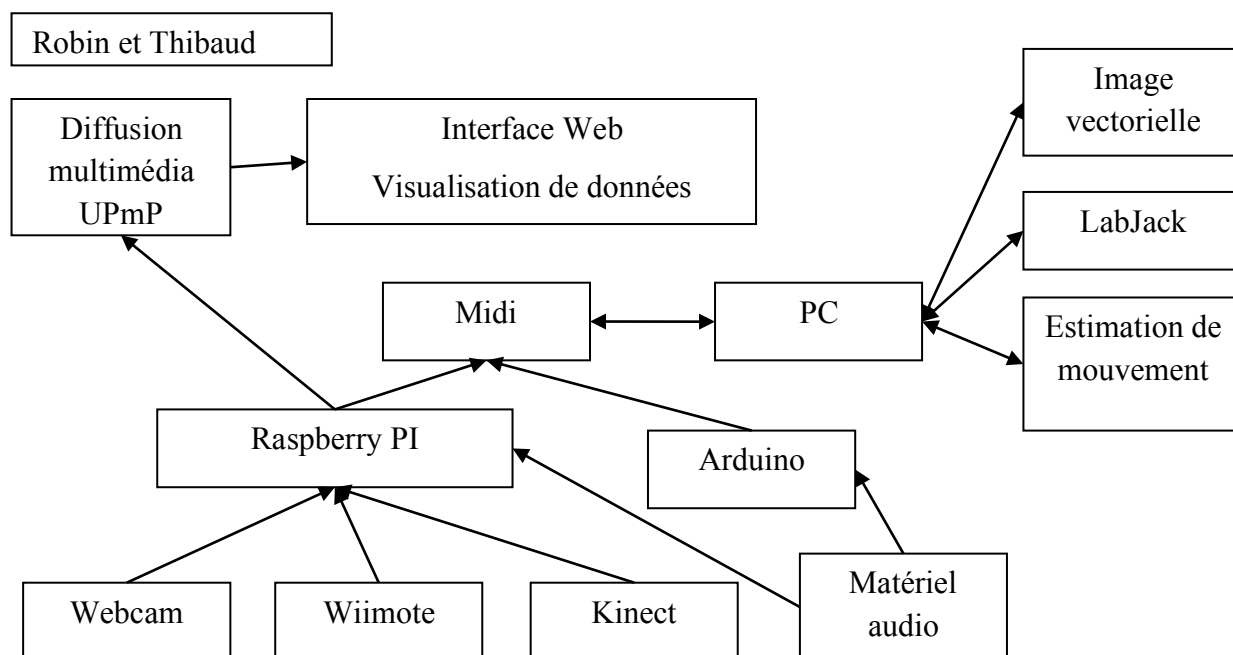
Après avoir mis en commun toutes nos idées, nous les avons regroupées par proximité de la façon suivante :



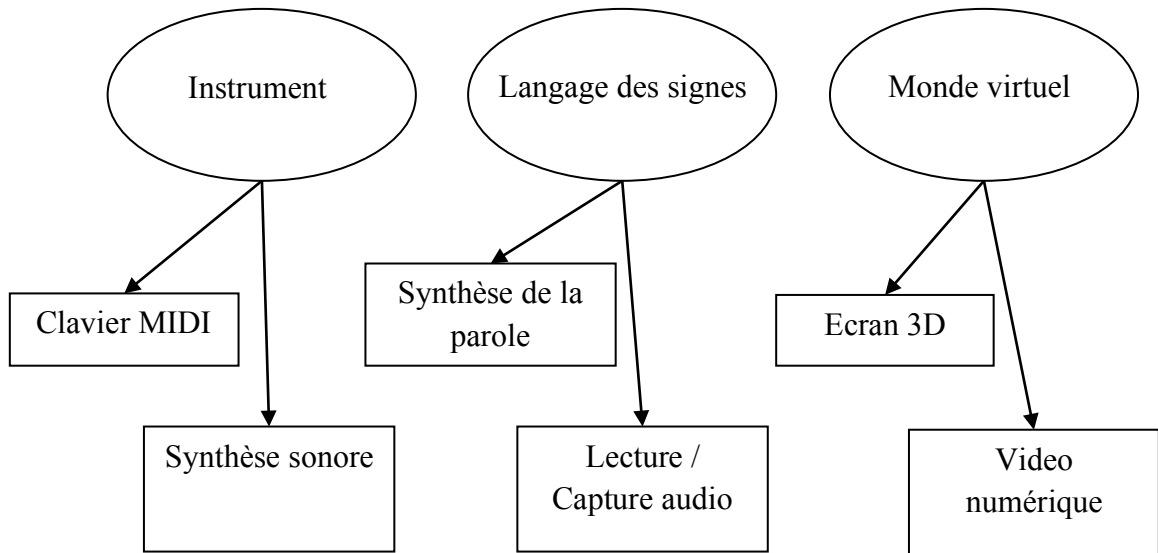
Répartition de technologies connues

Un ensemble de cartes décrivant des méthodes et des technologies nous ont été distribuées. Parmi celles que nous connaissons, nous avons dû trier les cartes correspondantes selon la proximité des domaines d'application de ces techniques, afin de les rapprocher des problèmes technologiques représentés par le projet PACT que nous allons devoir traiter.

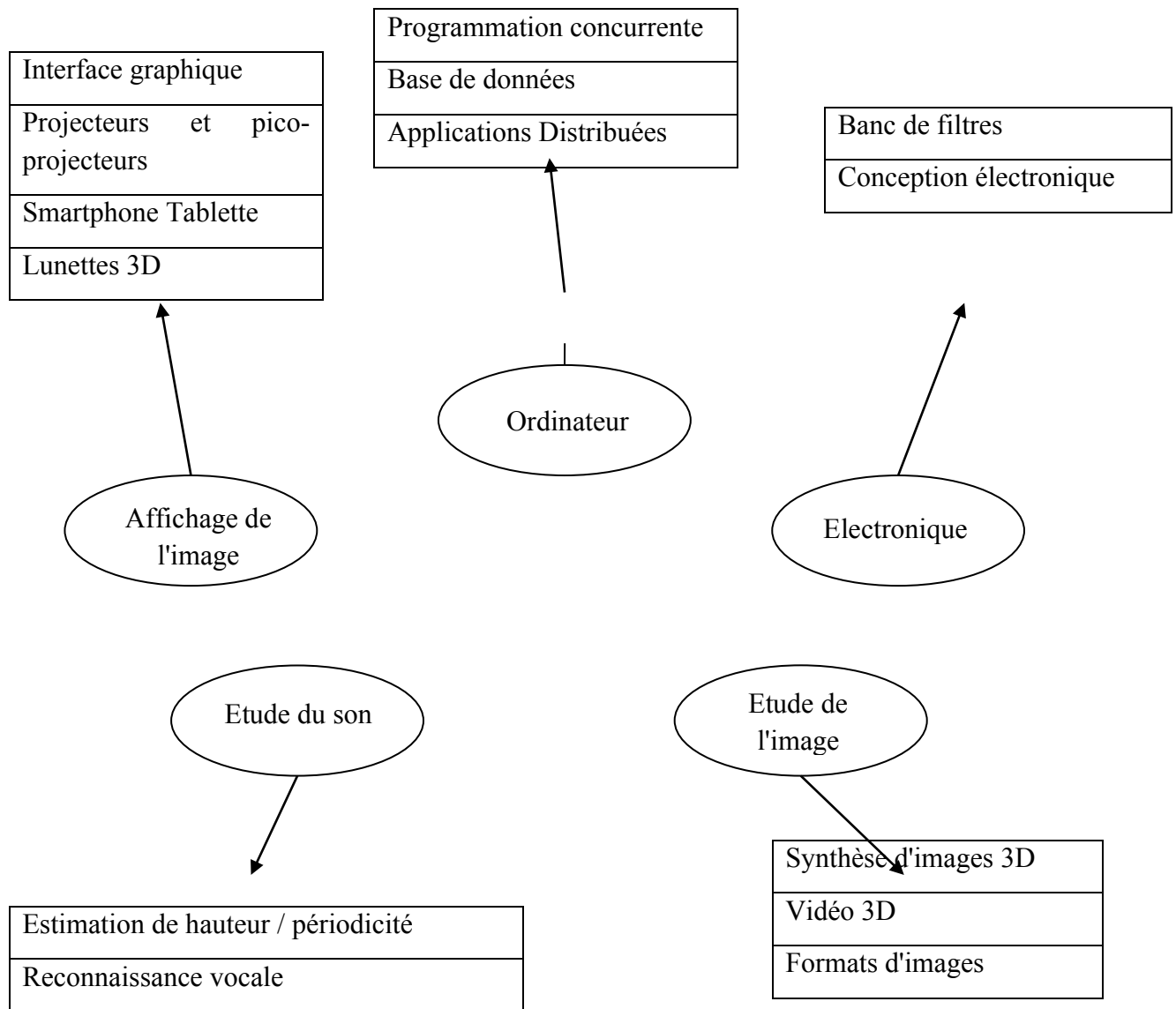
Trois regroupements différents ont donc été réalisés :



Stanislas et Alexandre



Khadija et Camille



Annexe D. Module Kinect

Module Kinect Squelette – Groupe 3.1

Encadrant: Jean Le Feuvre (jean.lefeuvre@telecom-paristech.fr, Poste 7169)

réalisé par : Robin SHIN et Thibault LEMAIRE

Volume estimé : 20 TH PACT + 10 TH perso – complexité normale

Descriptif

L'objectif de ce module est l'utilisation de la Kinect sous Windows en C++ via le SDK 1.8 de la Kinect. Le module permettra de récupérer les données du squelette et de les envoyer à Java via une méthode de communication simple.

Ressources dont le module dépend

Une machine récente sous Windows 7 ou 10 (PAS DE MACHINES VIRTUELLES)
SDK Kinect pour Windows ([site](#)), Visual Studio 2015 Community ([site](#))
La documentation Microsoft ([site](#))
La description du fonctionnement de l'algorithme de suivi du corps ([lien](#))

Objectifs d'apprentissage

- Principes de bases du fonctionnement de la Kinect
- Modélisation par squelette: notion de joint, de pose, d'invariance. Principes du fonctionnement de l'algorithme Microsoft.

Résultats attendus

PAN1

- expliquez simplement la Kinect: capteurs utilisés, données produites
- Quelles données sont disponibles pour le suivi de corps ? Comment ces données sont exploitées par votre prototype ?
- Avoir installé les logiciels sur une machine Windows.

PAN2

- Application test de suivi de l'ensemble du corps (les données du squelette peuvent être consultées via console ou dans un fichier) – **code C#1**
- Définition du format d'échange des données entre C et Java.
- Application client-serveur entre Java et C pour l'échange de données via sockets TCP – **code C#2 – Code Java#1**

PAN3

- Suivi du corps complet C->Java intégré dans le prototype allégé - **code C#3 – Code java #2**
- Possibilité de rejouer une séquence enregistrée
- **Documentation**
- Caractérisation de l'utilisation de la Kinect pour votre projet : les mouvements/gestes sont-

ils tous détectable comme prévu ? Quelles sont les conditions limites de détection dans le cadre de votre application (luminosité, distance au capteurs, types de gestes) ? Quels problèmes avez-vous pu contourner ? Quelles seraient vos pistes pour les autres ?

Annexe E. Module Classification

Module Classification – Groupe 3.1

Encadrant: Slim ESSID (slim.essid@telecom-paristech.fr)

Réalisé par : Thibaud Lemaire, Robin Shin et Karim Khandid

Volume estimé : ??? TH PACT + ? TH perso – complexité ???????????

Descriptif

L'objectif de ce module est de déterminer par une approche probabiliste les mouvements effectués par l'utilisateur. Le module vise à comparer les données provenant de la caméra Kinect (les points du squelette) à des mouvements préenregistrés dans une base de données.

Ressources dont le module dépend

- Aucune dépendance a priori mais possibilité d'exploiter une librairie JAVA (à choisir avec l'expert) pour développer le code de la classification.
- Définition mathématique d'une mesure, d'une distance, d'une similarité
- Une base de données (vectorielles) définie dans le projet du groupe

Objectifs d'apprentissage

- Notion en apprentissage automatique : distances, similarités, apprentissage, évaluation, validation croisée
- Compétence à développer : à l'issue de ce module vous aurez défini et programmé un classificateur kPPV en Java et vous saurez l'évaluer sur une base de données

Résultats attendus

PAN1 :

- **Savoir définir** les termes suivants : une distance, une similarité, apprentissage automatique, classification.
- **Savoir expliquer** en quoi cet outil de classification est a priori intéressant dans le projet du groupe.

PAN2 :

- *Livrable : **Etude bibliographique** sur les notions de distance et similarité*
- *Livrable : **Pseudo code** implémentant la classification par kPPV, en Java et le programme de test afférant.*

PAN3 :

- *Livrable : **code Java et premières mesures** de performance sur des données standards (par forcément en lien avec votre projet).*

PAN4 :

- *Livrable : **rapport** de performance, en situation d'usage.*

Annexe F. Détection du rythme

Module Détection du rythme – Groupe 3.1

Encadrant: Bertrand David (bertrand.david@telecom-paristech.fr)

Réalisé par : Stan Hannebelle, Khadija El attar

Volume estimé : ??? TH PACT + ? TH perso – complexité ????????????

Descriptif

L'objectif de ce module est de détecter le rythme et le tempo dans la danse de l'utilisateur ainsi que la richesse de sa danse

Ressources dont le module dépend

- ???

Objectifs d'apprentissage

- ??

Résultats attendus

- ??

Annexe G. Module Synthèse audio

Module Lecture, capture et traitement de fichiers audio – Groupe 3.1

Encadrant: Jean-Claude Dufourd (jean-claude.dufourd@telecom-paristech.fr)

Réalisé par : Khadija El Attar, Stan Hannebelle, Jamal Skouri

Volume estimé : ??? TH PACT + ??? TH perso – complexité ?????????

Descriptif

Ce module permet la lecture des fichiers audio en fonction des ordres du séquenceur musical. Le module génère et lit des boucles audio, gère le réglage du volume, et permet l'enregistrement de la musique générée dans un fichier

Ressources dont le module dépend

Bibliographie:

- <http://www.labbookpages.co.uk/audio/javaWavFiles.html>
- <http://commons.apache.org/math/> (la classe à utiliser pour la transformée de Fourier est `org.apache.commons.math3.transform.FastFourierTransformer`)
- <https://code.google.com/p/jmathplot/> ou <http://www.jfree.org/jfreechart/>
- <http://docs.oracle.com/javase/tutorial/sound/capturing.html>

Objectifs d'apprentissage

- Principe de la lecture audio en Java
- Création de fichiers sons
- Calage des boucles en rythme

Résultats attendus

- Connaissances : maîtrise du son en version informatique, pour des traitements courants
- Compétence : *à l'issue de ce module vous aurez écrit en Java un programme qui lit du son dans un fichier et le joue, capture du son et le stocke dans un fichier, calcule un son synthétique et le joue, visualise un son et son spectre.*

Livrable 1: programme qui lit un fichier son, le joue en boucle et en rythme

Livrable 2: programme qui enregistre dans un fichier

Module Android – Groupe 3.1

Encadrant: Jean-Claude Dufourd (poste 7733), bureau E509, jean-claude.dufourd@telecom-paristech.fr

réalisé par : Camille Demasson ,Khadija El Attar

Volume estimé : TH PACT + TH perso – complexité

Annexe I.

Descriptif

L'utilisation de smartphones ou de tablettes est devenue depuis quelques années presque incontournable. Parmi les différents systèmes d'exploitation permettant de contrôler ces smartphones et tablettes, le système Android représente près de la moitié des périphériques. La programmation des smartphones ou tablettes Android fait appel au langage Java, vu lors du cours INF103. Elle ne devrait pas poser de difficulté majeure, néanmoins une des difficultés souvent rencontrées est la prise en main de l'environnement de développement et des spécificités des applications Android. Ce module a pour vocation d'aider les élèves à surmonter cette difficulté.

Annexe J.

Ressources dont le module dépend

Le site de référence pour l'environnement de développement Android (bibliothèques, simulateur, environnement de développement, etc.) et pour l'auto-formation est le site <http://developer.android.com>.

Transparents du mini-cours:

[PACT-2014-Android-Mini-cours](#)

Une bonne introduction (en français) se trouve sur:

<http://openclassrooms.com/courses/creez-des-applications-pour-android>

Annexe K.

Objectifs d'apprentissage

Informatique : prendre en main l'environnement de développement Android, comprendre le fonctionnement d'une application Android, savoir lire et utiliser la documentation des API Android, savoir créer une application Android de base et utiliser certaines fonctionnalités avancées (affichage graphique, réseau, multimédia ...), savoir debugger une application

Annexe L.

Résultats attendus

PAN1 :

Démontrer que l'environnement de développement est en place et que des applications simples fonctionnent sur un émulateur

Expliquer le cycle de vie d'une application et la notion d'activité

PAN2 :

Réaliser une application Android sur émulateur mettant en oeuvre le cycle de vie, l'enchaînement de plusieurs activités, y compris en tâche de fond, et le lancement d'une autre application.

PAN3 :

Réaliser une application spécifique (à définir en fonction des besoins du projet). Cette application fonctionnera sur un smartphone ou une tablette et pourra mettre en oeuvre:

la communication via le réseau, par exemple selon le protocole HTTP,

un affichage graphique simple avec quelques éléments interactifs,

la lecture ou la capture de données multimédia (son, image, vidéo)

Décrire le fonctionnement et l'intégration du module dans le prototype allégé

PAN4 :

Analyser comment le module est intégré dans le prototype, quelles pistes d'améliorations seraient à envisager (performance, simplicité)

Module SES (Entretiens semi-directifs , Observations en situation naturelles) – Groupe 3.1

Encadrant: Julien Morel ()

Réalisé par : Jamal Skouri ,Stan Hannebelle ,Karim Khandid

Volume estimé : TH PACT + TH perso – complexité

Descriptif

Les entretiens semi-directifs peuvent être utiles en amont ou en aval du processus de conception. Ils consistent à recueillir la parole d'usagers (ou d'usagers potentiels) dans le but de comprendre leurs pratiques au sens large et surtout le sens qu'ils leur donnent. Ils consistent en une conversation guidée autour de plusieurs thèmes définis en amont, mais dont l'ordre et l'importance sont laissés ouverts. L'objectif des entretiens n'est pas de vérifier des hypothèses, ni de quantifier des résultats comparables, mais de faire émerger et de recueillir des discours (des éléments de vocabulaire, des définitions, des valeurs revendiquées, du sens très prosaïque, etc.) La méthode des entretiens débouche sur des préconisations précises dans le processus de conception. Celles-ci prennent la forme de contraintes ou d'éléments d'inspiration qui portent tant sur la manière de présenter la technologie développée que sur son possible ancrage dans des pratiques effectives. Le module consiste à réaliser et analyser une série d'entretiens. Il s'articule autour de quatre temps :

- élaboration d'un guide d'entretien complet
- choix de la population à interroger
- recueil de données (passation, enregistrement et retranscription des entretiens)
- rédaction d'un document de synthèse dédié au reste du groupe

Son encadrement consiste en une formation rapide aux techniques de l'entretien et à l'élaboration du guide. À la demande des élèves, un retour après chaque entretien est vivement conseillé.

Module Observation :

L'observation directe peut être utile en amont du processus de conception. Elle consiste à accéder, en situation naturelle à des attitudes, des comportements, des gestes, et plus généralement des cours d'action. Plus que des usages au sens strict, elle permet d'analyser des écologies complexes où de très nombreux paramètres sont en jeu. Sa mise en œuvre débouche sur des préconisations précises dans le processus de conception. Celles-ci

prennent la forme de contraintes (difficultés à prendre en compte, dimensions à éviter...) ou de points d'inspiration (meilleure compréhension des contextes d'usages possibles). Le module consiste à réaliser une série d'observations. Il s'articule autour de trois temps :

- élaboration d'une problématique claire et articulée au projet
- recueil de données pertinent et contrôlé
- rédaction d'un document de synthèse dédié au reste du groupe

Son encadrement consiste en une formation rapide aux techniques d'observation et en un suivi par retours réguliers (essentiels à la réussite du module) entre chaque séance d'observation.

L'utilisation de la vidéo (capture écran smartphone, lunettes-caméra, drone volant, multiplexer, etc.) sera encouragée.

Ressources dont le module dépend

Milieu de l'étude

Objectifs d'apprentissage

Résultats attendus

- compétence : *détachement des préjugés (sens commun, évidences...)*
- compétence : *développement des capacités d'écoute*
- compétence : *traitement et analyse de données qualitatives*
- livrable : guide d'entretien
- livrable : entretiens retranscrits (et leur enregistrement)
- livrable : document de synthèse des résultats destiné au groupe

livrable : explicitation dans la présentation finale du projet de l'apport de la méthode

- livrable : document de cadrage (problématique)
- livrable : recueil de données
- livrable : document de synthèse des résultats destiné au groupe

livrable : explicitation dans la présentation finale du projet de l'apport de la méthode

Annexe J. Module Intégration et tests

Module Intégration & test– Groupe 3.1

Encadrant: Soumia Dermouche (soumia.dermouche@telecom-paristech.fr)

Réalisé par : Alexandre Poinso ,Thibaud Lemaire ,Camille Demasson

Volume estimé : TH PACT + TH perso – complexité

Descriptif

L'objectif de ce module est de garantir l'interopérabilité des différents modules, de faire en sorte que tout fonctionne bien, et de faire les tests finaux.

Ressources dont le module dépend

Tous les membres du groupe
Tous les modules
Le dépôt GIT

Objectifs d'apprentissage

- Vérifier que chaque module implémente l'interface prévue (par compilation)
- Vérifier que chaque module a été testé (présence de tests dans le module)
- Créer les tests globaux, équivalents des méthodes de tests des modules au niveau du projet entier
- tests de fonctionnalité : ça fait ce qu'il faut
- tests de performance : ça va assez vite
- tests d'utilisabilité : c'est compréhensible par un utilisateur extérieur
- ...
- Appliquer un par un les cas de tests globaux prévus dans le plan de test:
- Faire des retours aux créateurs des modules sur les problèmes.
- Trouver des solutions pour les problèmes non spécifiques à un module.

Résultats attendus

PAN1

Description textuelle et schéma décrivant l'architecture du système
formalisme libre mais cohérent et légendé
Diagramme d'activité et/ou de séquence (temporel)
Faisant apparaître les acteurs et actions
Et les transitions avec leur(s) condition(s)
Description des interfaces entre modules (texte)
Dépot Git unique
Avec la bonne structure de dossier
Avec un commit par personne dans readme.txt

PAN2

Plan de test global

Liste des tests de fonctionnalité, de performance, d'utilisabilité, d'acceptabilité... (tous les types de test ne sont pas forcément pertinents pour votre projet)

Pour chaque test

contexte de test

liste des entrées à fournir

liste des sorties attendues, avec la méthode de validation (est-ce qu'on attend exactement ce résultat au bit près, ou dans une certaine marge d'erreur, ou un résultat de cette forme...)

Squelettes de toutes les méthodes principales

sous forme simulée

Fantôme : méthode vide utilisant `System.out.println` pour indiquer qu'elle est appelée.

Bouchon : méthode renvoyant toujours le même résultat.

Substitut : implémentation de la méthode très simplifiée.

PAN3 :

rapport sur l'état de l'intégration des modules (quel module, quelle version, quels problèmes) dans le proto allégé

résultat des tests (passés, échoués, résultats quantitatifs/qualitatifs sur les performances, l'acceptabilité ...) du proto allégé

Note: le rapport et les résultats se feront en utilisant les modules tels qu'implémentés, testés et livrés par les responsables des autres modules à une date à déterminer entre les responsables de modules. Il est acceptable qu'un livrable du module « test et intégration » ne contienne pas la dernière version d'un module (si ce dernier est en retard). Les retards seront cependant à justifier et feront l'objet d'une appréciation par le jury dans la notation de ces modules.

Note: l'encadrant va vérifier sur le dépôt Git que le code est documenté, lisible, maintenable, que tous les élèves ont contribué...

Annexe K Diagramme de GANTT

Diagramme de Gantt PACT

2 déc. 2015

<http://>

Chef de projet

Dates de début/fin du projet

28 sept. 2015 - 10 mai 2016

Avancée

0%

Tâches

88

Ressources

0

Tâches

2

Nom	Date de début	Date de fin
PAN 1	28/09/15	03/12/15
Communication orale et créative	28/09/15	28/09/15
Prise de contact et premier brainstorming	29/09/15	29/09/15
Envoi fiche d'identité du groupe	30/09/15	02/10/15
Idéation (convergence), préparation de 3 sujets	05/10/15	05/10/15
Foire aux experts	12/10/15	12/10/15
Foire aux modules	14/10/15	14/10/15
Choix du sujet et des modules associés	19/10/15	19/10/15
Scénario d'usage	19/10/15	29/10/15
Recherche de sujets/blocs élémentaires	15/10/15	17/10/15
Approfondissement du choix des modules – prise de rdv avec les experts	19/10/15	01/11/15
Travail encadré sur le scénario utilisateur par les experts SES.	05/11/15	05/11/15
Foire libre	09/11/15	09/11/15
Démarrage du travail en module	10/11/15	10/11/15
Initiation au Génie Logiciel	16/11/15	16/11/15
Avancement des modules et du GL – En autonomie	17/11/15	17/11/15
Génie Logiciel	24/11/15	24/11/15
Préparation du PAN1 – En autonomie	25/11/15	25/11/15
Finalisation du support et du discours pour le PAN1	26/11/15	03/12/15
PAN 2	08/12/15	20/01/16
Travail sur les modules – En autonomie	08/12/15	08/12/15
Travail sur les modules – En autonomie	14/12/15	14/12/15
Génie Logiciel	15/12/15	15/12/15
Avancement des modules	16/12/15	28/12/15

Tâches

Nom	Date de début	Date de fin
Avancement du projet	04/01/16	04/01/16
Préparation PAN2 et avancement du projet	11/01/16	11/01/16
Finalisation du support visuel et du discours pour PAN2	12/01/16	17/01/16
Soutenances PAN2	20/01/16	20/01/16
PAN 3	27/01/16	14/03/16
Débriefing PAN2 et avancement du projet	27/01/16	27/01/16
Avancement du projet	01/02/16	01/02/16
Avancement du projet	08/02/16	08/02/16
Avancement du projet	15/02/16	15/02/16
Avancement du groupe et préparation PAN3 = démo proto	29/02/16	29/02/16
Finalisation de la démo, du support visuel et du discours pour PAN3	01/03/16	13/03/16
Soutenances PAN3 = démo de proto allégé	14/03/16	14/03/16
PAN 4	15/03/16	09/05/16
Débriefing PAN3 et avancement du projet	15/03/16	15/03/16
Avancement du projet	21/03/16	21/03/16
Avancement du projet	30/03/16	30/03/16
Développement du proto final	04/04/16	04/04/16
Séance partage de connaissances et avancement du projet	05/04/16	05/04/16
Développement du proto final	11/04/16	12/04/16
Conception de la vidéo	18/04/16	18/04/16
Conception poster pour le stand + présentation de 4'	19/04/16	19/04/16
Prise de RDV avec les experts pour l'évaluation des modules	20/04/16	08/05/16
Soutenances PAN4	09/05/16	09/05/16
Présentations festives à tout le monde	09/05/16	09/05/16

Tâches

4

Nom	Date de début	Date de fin
Kinect - Thibaud LEMAIRE, Robin SHIN	28/09/15	14/03/16
Documentation Kinect	28/09/15	03/12/15
Installation des logiciels sous Windows	28/09/15	03/12/15
Application test de suivi de l'ensemble du corps	04/12/15	20/01/16
Définition du format d'échange entre C et Java	04/12/15	20/01/16
Application client-serveur entre Java et C pour l'échange de données via sockets TCP	04/12/15	20/01/16
Suivi du corps complet C -> Java intégré dans le prototype allégé	21/01/16	14/03/16
Possibilité de rejouer une séquence enregistrée	21/01/16	14/03/16
Caractérisation de l'utilisation de la Kinect pour le projet	21/01/16	14/03/16
Intégration & tests - Alexandre POINSON, Thibaud LEMAIRE, Camille DEMASSON	28/09/15	14/03/16
Description textuelle et schéma décrivant l'architecture du système	28/09/15	03/12/15
Construction du diagramme d'activité et/ou de séquence temporel	28/09/15	03/12/15
Description des interfaces entre modules	28/09/15	03/12/15
Liste des tests de fonctionnalité, de performance, d'utilisabilité, d'acceptabilité	04/12/15	20/01/16
Plan de test global	04/12/15	20/01/16
Squelette de toutes les méthodes principales sous forme simulée	04/12/15	20/01/16
Rapport sur l'état de l'intégration des modules dans le prototype allégé	27/01/16	14/03/16
Base de données	10/11/15	14/03/16
Application Android - Camille DEMASSON, Khadija EL ATTAR	28/09/15	09/05/16
Recherche sur l'environnement de développement et un émulateur	28/09/15	03/12/15
Recherche sur le cycle de vie d'une application et la notion d'activité	28/09/15	03/12/15
Réalisation d'une application Android sur émulateur	04/12/15	16/01/16
Réalisation d'une application spécifique à définir en fonction des besoins du projet	27/01/16	14/03/16

Tâches

Nom	Date de début	Date de fin
Description du fonctionnement et intégration du module dans le prototype allégé	27/01/16	14/03/16
Analyse de l'intégration du module dans le prototype et des pistes d'améliorations	15/03/16	09/05/16
Détection de rythme	10/11/15	14/03/16
Classification - Robin SHIN, Thibaud LEMAIRE, Karim KHANDID	28/09/15	09/05/16
Recherche sur les notions suivantes : distance, similarité, apprentissage automatique, classification	28/09/15	03/12/15
Recherche sur l'intérêt de l'outil de classification dans le projet	28/09/15	03/12/15
Etude bibliographique sur les notions de distance et similarité	04/12/15	16/01/16
Rédaction du pseudo-code implémentant la classification par kPPV, en Java et le programme de test afférant	04/12/15	16/01/16
Rédaction du code Java et premières mesures de performance sur des données standards	27/01/16	14/03/16
Rédaction du rapport de performance en situation d'usage	15/03/16	09/05/16
SES - Jamal SKOURI, Stan HANNEBELLE, Karim KHANDID	10/11/15	09/05/16
Entretiens semi-directifs	10/11/15	09/05/16
Observations en situations naturelles	10/11/15	09/05/16
Synthèse audio - Khadija EL ATTAR, Jamal SKOURI, Stan HANNEBELLE	10/11/15	09/05/16
Maitrise du son en version informatique	10/11/15	09/05/16
Codage d'un code Java qui lit un fichier son, le joue en boucle et en rythme	10/11/15	09/05/16
Codage d'un code Java qui enregistre dans un fichier les sons joués	10/11/15	09/05/16

Diagramme de Gantt

6

