

Projet MDMA - Rapport L2

Table des matières

1	Design Interface	3
1.1	Participants	3
1.2	Objectifs	3
1.3	Avancement	3
2	OpenCV Interface	3
2.1	Introduction	3
2.2	Conditions de fonctionnement	4
2.3	Algorithme développé	5
2.3.1	Structure de l'algorithme	5
2.4	Roadmap	6
3	Caractérisation des événements	6
3.1	Introduction	6
3.2	Description des zones	6
3.3	Implémentation	7
3.4	Affichage des zones	7
4	Workpackage Communication	7
4.1	Introduction	7
4.2	Définition d'une identité graphique	7
4.3	Communication numérique	8
4.3.1	Site Internet	8
4.3.2	Autre présence	10
4.4	Diffusion numérique	10
4.5	Bilan	10

Partie 1

Design Interface

1.1

Participants

- Hadrien Croubois (responsable d'équipe) ;
- Luc Rocher

1.2

Objectifs

Le *workpackage* "Design Interface" a pour objectif de mettre au point et d'implémenter l'interface ainsi que de penser à l'interconnexion de l'interface avec l'ensemble des composants du logiciel. Il s'occupe en plus de la gestion des configurations décrites par l'utilisateur, qu'il s'agisse de leurs paramétrages via les différentes fenêtres ou de la partie chargement/enregistrement. La connaissance de l'état d'avancement des différents *workpackages* permet de définir efficacement les formats de configuration ainsi que de penser les flux de données nécessaires à l'affichage au cœur de la structure du programme.

Le *workpackage* "Design Interface" anticipe à ce titre une part de l'intégration.

1.3

Avancement

L'interface en elle-même est écrite à l'aide du designer de fenêtre de *QtCreator*¹.

La fenêtre principale permet, en plus de la visualisation de l'image acquise par la webcam de charger et sauvegarder différentes configurations ainsi que d'avoir un résumé des différentes zones définies. Ces zones peuvent être créées en cliquant sur la vidéo, modifiées et supprimées.

L'ouverture d'une nouvelle configuration ou la fermeture du programme alors que la configuration actuelle a été récemment modifiée demande à l'utilisateur de sauvegarder la configuration actuelle.

Un bouton "Calibrate" lancera une procédure de calibration, de même que le bouton "Configure" ouvre une fenêtre permettant de choisir différentes options (caméra si plusieurs sont disponibles, port midi ...).

Un bouton Run lance le programme. Une fois lancé, la configuration n'est plus éditée (les différents boutons sont désactivés). Seul le bouton "Run", renommé en "Stop", reste actif et permet d'arrêter le programme.

Partie 2

OpenCV Interface

2.1

Introduction

La tâche prise en charge initialement par les contributeurs du *workpackage* consistait en la création du module qui capturerait l'image de la caméra, loca-

1. QtCreator - <http://doc.qt.digia.com/qtcreator/index.html>

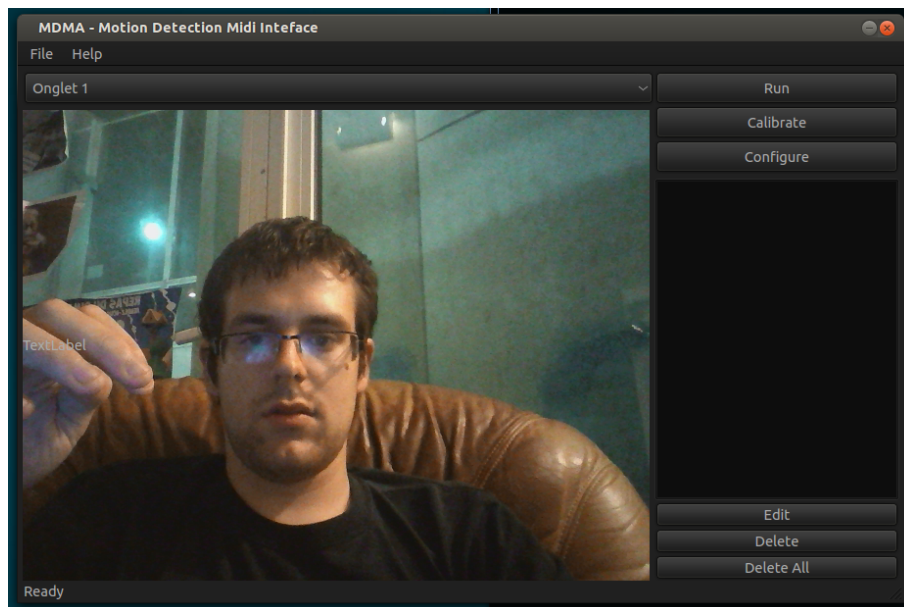


FIGURE 1 – La fenêtre principale de l'interface

liserait les mains sur cette image et transmettrait la totalité des informations : la position des mains, leur vitesse, leur accélération et la forme (fermée ou ouverte). La spécification initiale du système sous-entendait la possibilité que le système fonctionne sans utiliser de matériel onéreux.

2.2

Conditions de fonctionnement

Pendant le développement, il était entendu que la reconnaissance des mains se ferait sans avoir recours à des marqueurs et sans l'utilisation de matériel spécial (comme par exemple l'outil Microsoft Kinect). Après avoir considéré certaines contraintes pour assurer la reconnaissance des mains, l'équipe de ce workpackage a décidé d'utiliser des gants noirs. Les tests ont montré que la reconnaissance des mains avec les gants est stable sous les quelques conditions suivantes :

- Le fond ne contient pas d'objets foncés.
- Le vêtement d'utilisateur est notablement plus clair que les gants, qui doivent être très foncés (noirs).
- L'éclairage doit être suffisant.

Du point de vue logiciel, le module nécessite que la librairie OpenCV soit installée sur le système.

Comme extension du projet, nous envisageons de développer une version de notre logiciel fonctionnant avec une Kinect. La version ne sera qu'optionnelle tant qu'elle demandera les dépenses supplémentaires de part de l'utilisateur. Malgré ce défaut le développement de la version est bien raisonnable tant qu'il assure la reconnaissance grâce à l'utilisation des matériels dont la prédestination est le tracking des objets.

2.3

Algorithme développé

Vers ce moment le résultat est l'algorithme développé et implémenté avec l'utilisation de librairie OpenCV. L'implémentation développée d'une manière qui lui permet d'être plate-forme-indépendante.

Structure de l'algorithme

- Calibration du système. L'utilisateur place les mains dans les zones spéciales. Après cela le système fixe L' égal éclairage du point le moins éclairé dehors de ces zones. $L = L' - \Delta$ devient le seuil critique de luminosité. Après cette étape le main boucle infini du fonctionnement se déroule.
- Commencement du boucle. Capture de l'image.
- Conversion RGB \rightarrow Grayscale.
- Création du masque. Les 1 prennent place de tous les pixels dont la luminosité est moins que L .
- Masquage des pixels de la zone morte dédié pour le corps de l'utilisateur. Les pixels de la zone se rendent égaux 0.
- L'érosion de l'image.

Définition de l'érosion. L'érosion est l'une des deux opérations fondamentales du traitement d'image morphologique. Soit A une image binaire et B un élément de structure. L'érosion de A par B est donné par : $A \odot B = \{z | (B)_z \subset A\}$.

Pour l'élément structurant dans l'implémentation nous utilisons ellipse 3×3 .

0	1	0
1	1	1
0	1	0

TABLE 1 – Élément structurant : ellipse 3×3

L'érosion est utilisée pour la filtration du bruit : petites particules blanches qui peuvent apparaître sur le masque par hasard. L'érosion est appliquée au masque 2 fois.

- La dilatation de l'image.
La dilatation a pour le but compenser l'utilisation de l'érosion et simplifier les frontières des objets. La dilatation est définie comme $A \oplus B = \cup_{b \in B} A_b$. Pour la dilatation le même élément structurant que pour l'érosion dans l'étape précédente.
- La recherche des contours. La recherche des contours est faite avec l'utilisation de l'algorithme de Teh-Chin².
- La construction de l'enveloppe convexe. Les points des contours sont divisés en deux ensembles selon la partie de l'image (gauche ou droite) à laquelle ils appartiennent. Pour chaque ensemble on construit une enveloppe convexe avec l'algorithme de Sklansky³.

2. Teh, C.H. and Chin, R.T., On the Detection of Dominant Points on Digital Curve. PAMI 11 8, pp 859-872 (1989)

3. Sklansky, J., Finding the Convex Hull of a Simple Polygon. pp 79-83 (1982)

- Le moyen des coordonnées de chaque enveloppe est considéré comme la coordonnée de la main. Gardant l'histoire des positions des mains on peut calculer la vitesse et l'accélération.
- On recommence la boucle.

2.4

Roadmap

Comme l'algorithme est développé et testé il ne reste que l'intégration avec workpackage Gestion des événements. C'est à dire il reste créer l'objet actif qui offre l'interface au module de gestion des événements.

Partie 3

Caractérisation des événements

3.1

Introduction

Le workpackage de caractérisation des événements se charge du déclenchement des signaux MIDI fournis par l'interface MIDI lorsque des événements sont détectés dans des zones décrites par l'utilisateur et de l'affichage en surimpression des zones activées. Les spécifications nous donnent les types d'événements qu'il faut pouvoir détecter et quels types de zones on doit disposer et l'interface MIDI nous fournit ce qu'on doit envoyer. Il s'agit donc d'implémenter la détection d'évènement, le déclenchement des signaux et le dessins des zones.

3.2

Description des zones

Une zones est décrites initialement par sa position, sa taille, son type (fadder, pad ou segment) et, pour chaque événement, un ensemble de signaux MIDI (éventuellement vide) qui doivent être déclenché lorsque cette événement est détecté. Les événements que l'on souhaite repéré dans une zone sont l'ouverture et fermeture de main, le déplacement de main, le franchissement pour un segment et l'entrée et la sortie pour un pad. (les pads XY seront gérés comme des fadders à axes doubles)

Pour détecter un événement, on dispose de deux positions de la main, l'actuelle et la précédente et de l'état de la main, ouverte ou fermée. De ces informations on détecte directement l'entrée/sortie d'une zone, le franchissement d'un segment. Pour les autres événements, il nous faut également l'état de la zone :

- active ou non pour lancer ou arrêter un clip
- l'ouverture de la main à l'entrée dans la zone pour les fadders, une main qui rentre fermée dans un fadder ne doit pas changer sa valeur.

Une fois un événement détecté, on envoie les signaux correspondant à l'interface midi.

3.3

Implémentation

La détection ET le déclenchement de signaux sont réalisés par les zones. La gestion des événements consiste à envoyer à chaque zone/segment les informations suffisantes relatives à la main. Il s'agit donc d'une boucle disposant de la liste des zones (fournis par la configuration) qui la parcourt en appelant pour chaque zone la détection.

3.4

Affichage des zones

Cette partie n'a pas encore été codé car est le résultat de notre dernière réunion. Les zones et les segments seront dessinés sur un Label Qt puis sera superposé à l'image de la caméra avec de la transparence. Il s'agit de montrer quelles zones sont actives et donc d'offrir un support visuel à l'utilisateur.

Partie 4

Workpackage Communication

4.1

Introduction

Le *workpackage* « communication » est composé de trois personnes : Luc Rocher (responsable d'équipe), Timothée Bernard & Hadrien Croubois.

Il s'occupe de la communication du projet sur deux échelles, en externe et en interne. La communication externe concerne l'ensemble des liens entre *MDMA* et les acteurs extérieurs (que ce soit par voie numérique – site Web, courriels – ou par voie orale). La communication interne est chargée de gérer les discussions dans l'équipe, ainsi qu'avec les responsables du projet et du département.

4.2

Définition d'une identité graphique

Pour débiter une bonne communication, il faut d'abord réfléchir à l'image que l'on veut donner au projet. MDMA a pour but de développer un logiciel innovant : l'interface homme-machine se veut délibérément moderne, sans clavier ni souris l'utilisateur peut commander son ordinateur pour produire des signaux musicaux. Mais il s'inscrit dans une tendance alternative, celle de la musique électronique et des synthétiseurs sonores. Nous avons cherché à construire une identité assez *rétro* mais véhiculant la nouveauté du projet.

Nous sommes parti d'un ensemble de couleurs très vivantes – du jaune au rouge – assez agressives :



FIGURE 2 – Gamme de couleurs utilisées pour la charte graphique

Avec ces couleurs, nous avons pu réfléchir à un logotype. Plusieurs pistes étant possible, celle d'un texte uniquement (« MDMA ») assez minimaliste a été privilégié, avec une police très *pixelisée*. Trois propositions ont été réalisées pour affiner le logo :



FIGURE 3 – Trois propositions de logotype

Le premier graphique a été sélectionné par l'équipe, et rapidement intégré au site Internet.

4.3

Communication numérique

Site Internet

Afin d'obtenir un minimum de visibilité, le projet avait besoin d'une présence sur Internet. Nous avons donc d'abord travaillé à mettre en place un site Web. Nous avons travaillé sur une interface assez minimaliste, fondée sur les couleurs de la gamme représentée à la figure 2.

Le site Internet (<http://graal.ens-lyon.fr/mdma/>) contient différentes informations, dont :

- une présentation du projet, minimale sur la première page, puis détaillée par la suite ;
- une description complète de chaque *workpackage* ainsi que leurs avancées et découvertes au fil du temps ;
- une page personnelle pour chaque membre, avec la possibilité de nous contacter.

Le site est fondé sur Wordpress – un moteur dynamique – qui nous permet de gérer des commentaires pour chaque page, une première page dynamique avec la liste des derniers articles, des flux RSS... Par ailleurs, un **travail de traduction** a été réalisé sur presque l'intégralité du site, pour obtenir un site entièrement bilingue, en français et en anglais. Un menu en haut du site contient des boutons avec icônes pour sélectionner la langue ; par ailleurs, selon la langue du navigateur, la langue optimale est automatiquement choisie.



FIGURE 4 – Capture d'écran du site (version mobile)

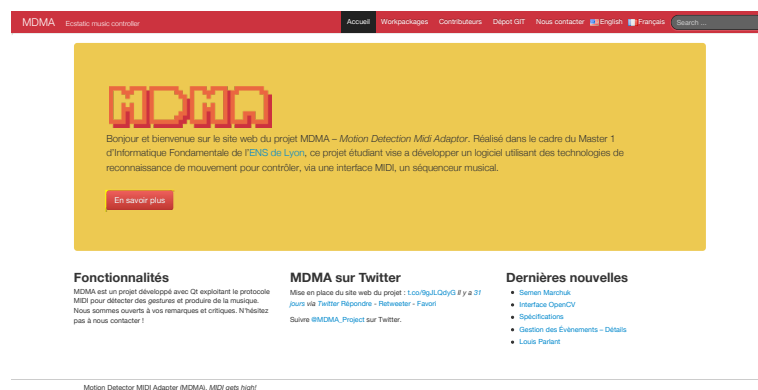


FIGURE 5 – Capture d’écran du site du projet

Autre présence

Le projet dispose aussi d’un page sur Facebook (<http://www.facebook.com/mdma.project>) et d’un compte Twitter. Ces espaces ne sont pas très actifs, mais pourront le devenir dès que le projet disposera d’un logiciel fonctionnel et pouvant être diffusé.

4.4

Diffusion numérique

L’équipe « communication » se charge du lien entre le projet et l’équipe pédagogique, par exemple pour des questions de budget (le financement récent d’un Microsoft Kinect par exemple) ; il a aussi pour vocation de diffuser notre projet à des musiciens et des personnes intéressées par la musique électronique. Nous chercherons ainsi des utilisateurs capable de tester MDMA et de produire des retours en situation réelle. Cette axe là pourra se développer dans les semaines qui viennent, avec la finalisation du logiciel.

4.5

Bilan

À l’orée de la moitié du projet, la communication a réalisé une grande partie du travail nécessaire. Il reste encore à rajouter du contenu sur le site, à favoriser les échanges avec les professionnels de la musique, et pourquoi pas à étudier un volet *presse*.