# Rapport Projet Kinect

## Liste des gestes reconnus

* Balayage Droit (main droite)
* Balayage Gauche (main gauche)
* Balayage Haut (2 mains)
* Coup de Poing Droit Avant
* Course
* Kage Bunshin

## Solutions Algorithmiques pour les Détections

Pour chacun de ces gestes, nous avons choisi de prendre comme « distance de référence » la taille du torse (toutes les distances seront comparées à cette valeur, ou une portion de cette valeur)

### Balayage Droit

Nous analysons le mouvement en 3 états :

1. La main est proche du torse (distance main/épaule < 1/3 du torse)
2. La main est « au milieu » (entre 1/3 du torse et 2/3 du torse)
3. La main est éloignée du torse (>2/3)

Lorsque l’on atteint l’état 3 et que l’on revient à l’état 2 ou 1, le mouvement est détecté.

### Balayage Gauche

Même chose que pour le balayage droit, sauf qu’on vérifie en plus que la main est à gauche de l’épaule droite

### Balayage Haut

Ici seulement 2 états : mains basses, puis mains hautes, puis si on remet les mains vers le bas, le mouvement est détecté (donc on joue sur la position y des mains et des épaules).

### Coup De Poing Droit Avant

Toujours le même principe avec les 3 états (on regarde la position en z de la main et de l’épaule), sauf qu’on ajoute en plus une condition sur la composante en y de ces membres, dans le sens où la main et l’épaule doivent rester à peu près au même niveau pendant tout le coup. Nous avons ajouté cette condition pour éviter que les mouvements de la course ne viennent générer des coups de poing.

### Course

Nous avons 3 états :

1. La main droite est levée et la main gauche est basse
2. Les 2 mains sont à peu près au même niveau y
3. La main gauche est levée et la main droite est basse

Pour détecter cela, on compare la composante en y de la main et du coude (si yMain – yCoude est positif, la main est au-dessus du coude, et inversement si c’est négatif).

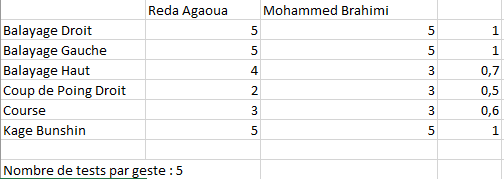
### Kage Bushin

Nous avons 2 états :

1. Les mains proches du corps
2. Les mains forment un « sceau » de ninja (l’avant-bras gauche forme un angle droit avec l’avant-bras droit, et le bras gauche est derrière le bras droit).

Pour cela nous avons simplement vérifié que le coude gauche et la main gauche étaient à peu près au même niveau y, que le coude droit et la main droite étaient à peu près au même niveau x, enfin que la main droite et la main gauche se trouvent à peu près au même niveau (on utilise pour ça une marge).

## Résultats obtenus



Les balayages droits et gauches sont détectés la plupart du temps. En revanche les balayages hauts, coups de poings et course sont détectés avec bien moins de précision. Le problème réside dans le passage vers l’état intermédiaire : les seuils de distances sont peut-être trop imprécis pour permettre de passer vers cet état fréquemment, et donc diminuent la fréquence de détection de ces mouvements. Nous avions ajouté cet état intermédiaire pour donner plus de précision à la définition du mouvement (qui ne se fait pas « d’un seul coup »).

### Manuel d’utilisation du package

Le package est composé d’un prefab contenant un différents gameObjects portant les scripts nécessaires à la kinect, ainsi que nos scripts ModelController et les mouvements que nous voulons détecter. On peut aussi y trouver le script Movement.cs. Pour ajouter un mouvement, il suffit d’hériter de cette classe (Movement), d’ajouter les variables contenant la position des éléments traqués, puis de réécrire les fonctions NewStateUpdate() et StateTransition(). Ensuite, on ajoute ce script au gameObject contenant le MovementController, et on met à jour ce MovementController pour qu’il agisse lorsqu’il détecte le nouveau mouvement.