

Rapport de projet IEC

# Reconnaissance de gestes via Kinect

Raphaël Santos  
Eva Stéphane

*28 Octobre 2018*

## Gestes reconnus

L'ensemble des gestes reconnus par notre application est :

- balayage de la main droite vers la droite et retour à la position d'origine (swipe right)
- balayage de la main droite vers la gauche et retour à la position d'origine (swipe left)
- balayage des deux mains en même temps vers le haut puis retour (swipe up)
- coup de poing de la main droite vers l'avant (punch)
- un geste de notre choix : deux mains levées en même temps, en diagonale (Praise the Sun)



Swipe up



Punch



Swipe Left



Praise the Sun



Swipe right

Bien que implémenté dans le code et les interfaces, nous n'avons pas réussi à faire fonctionner le geste Run, probablement à cause de seuils de vérification trop contraignants et une décomposition du mouvement non-adapté.

## Solutions algorithmiques choisies pour la détection

Avant de procéder à la reconnaissance des gestes en elle-même, nous avons fait la distinction entre les gestes statiques, qui sont invariants dans le temps et dans l'espace, et les gestes dynamiques, variants dans le temps et l'espace. En particulier, les gestes dynamiques peuvent être vus comme une suite de gestes statiques. Par exemple, le geste

*balayage de la main droite vers la droite puis retour* correspond au geste statique *main droite tendue vers la droite* suivi du geste *main droite près du corps*.

Un geste dynamique correspond donc à un enchaînement unique de gestes statiques. Pour cette raison, nous avons adopté le modèle des machines à états pour reconnaître les gestes effectués par l'utilisateur.

Chaque geste à reconnaître est défini par une liste de gestes statiques. Dès que l'utilisateur effectue un mouvement qui est reconnu par l'application, ce mouvement est ajouté à la liste des gestes détectés. À chaque nouvel ajout dans cette liste, celle-ci est comparée à l'ensemble des listes de gestes statiques constituant les gestes à reconnaître. Dès qu'un geste est détecté, l'utilisateur est averti par des retours visuels.

La reconnaissance étant conçue pour que l'utilisateur soit assis, elle est basée sur le mouvement des deux mains. La machine à états fonctionne donc avec des couples de gestes : les positions des deux mains sont récupérées à chaque frame, puis le couple formé par le geste reconnu pour chaque main est ajouté à la liste des gestes détectés.

Comme certains gestes ne sont basés que sur le mouvement d'une main, nous avons ajouté un état possible, "*none*", indiquant que le geste de l'autre main ne doit pas être pris en compte.

Nous avons basé la reconnaissance sur la position des articulations de l'utilisateur. Ainsi, les gestes statiques qui nous servent à construire les gestes dynamiques consistent à vérifier qu'une articulation est proche d'une autre ou, au contraire, est éloignée d'une certaine distance. Cette modélisation permet de facilement reconnaître les translations puisqu'il suffit d'ajouter un état pour chaque point clef de la translation. Par exemple, le geste *balayage des deux mains vers le haut* est décomposé de la façon suivante :

- les deux mains sont près du corps
- les deux mains sont au dessus de la tête
- les deux mains sont près du corps

## Tests et résultats

La reconnaissance a été testée sur 3 personnes, chacune devant exécuter cinq fois un geste demandé avant de passer au suivant, et ne voyant pas l'écran pour ne pas être influencée par ce que renvoyait l'interface utilisateur.

On notera que l'on ne retiendra pas les résultats du troisième utilisateur, car ce dernier était de grande taille (1m95). En effet, nos tests lors du développement ont été réalisés sur des personnes de taille moyenne ou petite, et notre reconnaissance est basée sur la distance entre les articulations de l'utilisateur. Le cas des personnes de grande taille n'avait donc pas été géré et nous ne nous en sommes aperçus qu'au moment de tests ultérieurs, soutenance comprise.

Les résultats des tests sont les suivants :

Gestes	Utilisateur 1	Utilisateur 2	Utilisateur 3 (non retenu)	Total
Swipe right	60 %	60 %	20 %	60 %
Swipe left	40 %	60 %	0 %	50 %
Swipe up	20 %	20 %	0 %	20 %
Punch	60 %	80 %	20 %	70 %
Praise the sun	80 %	100 %	40 %	90 %

Les résultats sont satisfaisants pour le Punch et le geste personnel, pourraient être améliorés pour le Swipe Right et le Swipe Left, et est largement insuffisant pour le Swipe Up. Il est possible que les conditions sur les deux premiers Swipe soient trop contraignantes. Nous supposons que le très faible score du Swipe Up viennent d'une contradiction avec le mouvement personnel, qui l'occulterait en partie. Enfin, le Punch est relativement élevé car il s'appuie sur une étude sur la profondeur. En conclusion, il faudrait que nous trouvions des seuils de vérification plus flexibles pour les différents swipe, et réussir à différencier un peu plus le Swipe Up et le Praise The Sun.

## Manuel d'utilisation

L'utilisation du package se fait de la manière suivant:

- Importer le "KinectOverlay" via le menu *Assets > Import Packages > Custom Packages*.
- Une fois que cela est fait, trois dossiers apparaîtront. Dans le dossier *Scripts*, vous trouverez le code utilisé pour la reconnaissance de gestes. Dans le dossier *Prefabs*, l'ensemble des prefabs utilisables dans le cadre du *Kinect Overlay*, et enfin, le dossier *Examples* propose des scènes d'illustration, accompagnées de leurs scripts et interfaces..
- Pour déployer la reconnaissance de geste dans une scène, glisser simplement le prefab *CustomKinectController* dans la scène, et vous aurez accès à la reconnaissance. Il générera lui-même le *Kinect Manager* et les autres composants nécessaire à son fonctionnement.
- Pour pouvoir visualiser l'utilisateur, glisser le prefab *SphereMan* dans la scène. S'il détecte un *KinectManager*, il s'animera lui-même pour afficher la position de l'utilisateur dans l'espace.

## Ajout d'un nouveau geste

L'ajout de geste n'a pas été rendu possible via l'inspecteur de Unity. Pour ajouter un geste qui ne figure pas dans les gestes proposés, il faut le créer dans le script *ComplexGestures*. Ce script construit entièrement un geste dynamique, à savoir :

- Créer la liste des couples de gestes statiques qui constituent le geste à reconnaître ;
- Créé, à l'aide de cette liste, un objet *Gesture* correspondant au geste à reconnaître ;
- Ajouter ce geste à la liste des gestes reconnaissables par l'application.

Il faut ensuite ajouter le cas où ce geste est reconnu dans la méthode *CompareGestureSequenceWithAllComplexGestures()* du script *RecognizeGesture*.

Si le geste que l'on souhaite ajouter comporte des gestes statiques qui ne sont pas implémentés, il est possible d'ajouter ses propres gestes statiques. Pour cela, il faut d'abord ajouter l'identifiant que l'on va associer au geste statique dans l'enum *GestureId* du script *Gesture*. Ensuite, la reconnaissance du geste simple est implémentée dans le script *SimpleGesture*, et appelée dans le script *RecognizeGesture*, dans la méthode *CheckJointsPositions()*.