Guillaume GONTHIER, Louis-Baptiste TRAILIN

Projet Kinect ~ Documentation

1/ Gestes reconnus

Balayage à droite

- En partant avec la main droite vers l'avant, au niveau du nombril, effectuer un geste vers la droite, puis revenir à la position initiale, le tout assez rapidement (moins d'une seconde)

Balayage à gauche

- En partant avec la main droite vers l'avant, au niveau du nombril, effectuer un geste vers la gauche, puis revenir à la position initiale, le tout assez rapidement (moins d'une seconde)

Balayage en haut

- En partant avec les deux mains posées sur les genoux, les lever simultanément vers le haut, et les descendre de manière synchrone, le tout en un peu plus d'une seconde

Coup de poing

- En gardant toujours le poing droit à la même hauteur (au niveau du nombril), ainsi qu'en gardant le coude et le poignet à l'horizontal, sur le même niveau, effectuer un geste d'aller-retour rapide vers l'avant

Simulation de course

- En gardant toujours les poings à la même hauteur (au niveau du nombril), ainsi qu'en gardant les coudes et les poignets à l'horizontal, effectuer au moins 4 gestes d'aller-retour rapides vers l'avant (deux avec le bras gauche, deux avec le bras droit), en alternant un bras en avant, et l'autre en arrière, le tout de manière continue (il est important de noter que la détection prend un peu plus de temps que pour les autres tests)

Salut militaire

- En partant avec la main droite posée sur les genoux, la lever verticalement jusqu'au niveau de la tête, en ayant, une fois arrivé, le coude au niveau de l'épaule

2/ Solutions algorithmiques pour la détection

Afin de détecter les mouvements, nous avons mis en place un système de buffer de taille réglable, avec un échantillonnage dont on peut choisir la fréquence. Pour la plupart de cas, nous avons défini un buffer d'une seconde et un échantillonnage à 10 Hz (soit 10 mesures de 100ms stockées). Nous travaillons sur deux tableaux, possédant des informations différentes : nous stockons dans le premier la dernière position connue du membre (en récupérant la composante qui nous intéresse), et dans le second, la différence entre deux positions successives. Nous utilisons des tableaux cycliques, de taille fixe.

Lors du test de détection, nous parcourons les tableaux en partant du dernier élément enregistré, afin de « revenir dans le passé », voir l'évolution de la position, et donc détecter le mouvement à l'envers. Avec ce système de buffer, nous pouvons détecter des mouvements tant que leur durée est comprise dans un certain intervalle, délimité par la taille du buffer et l'échantillonnage.

Pour connaître la position d'un membre, nous associons le GameObject du membre du stickman au script. Si pendant du test le mouvement est détecté, nous mettons un booléen à vrai, et un cooldown local d'une seconde s'active. Ce dernier a pour but de laisser le temps aux autres scripts de savoir qu'un mouvement a été détecté.

Nous avons aussi mis un cooldown global à tous les scripts de détection afin d'éviter d'avoir deux gestes détectés en même temps. Par conséquent il faut attendre une seconde après qu'un geste ait été reconnu, avant que le suivant ne soit détecté.

On notera enfin le cas particulier de la course et du coup de poing, car ces deux gestes entrent en conflit. Pour le résoudre, nous avons mis ces deux gestes dans le même script, car leur détection passe non seulement par les mêmes buffers, mais en plus, on peut voir une course comme un enchaînement de coups de poing.

3/ Résultats des tests et discussion

Gestes	Balayage à droite	Balayage à gauche	Balayage en haut	Coup de poing	Simulation de course	Salut militaire
Alexis Giraudet						
	1	1	1	1	0	1
	1	1	1	1	0	1
	1	1	1	1	0	0
	1	1	1	0	0	1
Taux de détection	100%	100%	100%	75%	0%	75%
Lucie Lebon						
	1	0	1	1	1	0
	1	1	1	1	1	0
	1	0	1	0	1	0
	1	0	1	1	0	0
Taux de détection	100%	25%	100%	75%	75%	0%
Moyenne par geste	100%	63%	100%	75%	38%	38%
Moyenne globale	69%					

On remarque que la plupart des gestes sont détectés de manière fiable, avec une moyenne totale de 69% de gestes correctement identifiés. On notera cependant de fortes disparités entre certains mouvements, et, chose plus surprenante, des gestes très bien détectés pour un testeur, s'avèrent totalement raté pour un autre.

Notre hypothèse est que, dans le cas du salut militaire, il n'a pas bien fonctionné pour un testeur car la Kinect détectait le coude droit trop haut, ce que perturbait le test de détection. Concernant la course, elle n'a pas correctement fonctionnée pour un testeur car la Kinect détectait les bras au-dessus des épaules, alors même qu'ils étaient en réalité au niveau du nombril du testeur. Le balayage à gauche n'a pas correctement fonctionné car le mouvement effectué n'était pas assez ample.

4/ Manuel d'utilisation du package

Concernant la manière dont est organisée notre scène, nous avons mis des textes, servant d'indicateurs pour savoir quel mouvement est détecté. Ces derniers sont colorés en rouge par défaut, et passent au vert quand un geste est identifié, en récupérant le booléen du script lié au geste en question.

Dans le menu principal, il y a deux cubes rouges, qui représentent les boutons quitter l'application et démarrer (respectivement sortir) du mode détection libre. Il suffit de rester dessus avec la main droite du stickman pendant 3 secondes pour utiliser la fonction du bouton. Il est important de noter qu'on peut aussi utiliser le clavier : pour lancer la détection, il faut appuyer sur la barre espace, Echap sert quant à lui à revenir au menu principal, tandis que la touche entrée permet de quitter l'application.

Pour utiliser notre prefab fait sous Unity, il faut mettre le prefab de Kinect dans la scène sur laquelle on veut travailler. Il suffit ensuite d'associer le SkeletonWrapper au sous-objet RainbowMan, et le prefab fonctionnera sans problème.

Concernant les différents paramètres des scripts, nous avons mis comme valeur par défaut (on notera que generalCooldown est toujours à 1.0, et qu'il faut toujours associer à la variable GeneralCooldown le script generalCooldown) :

- SwippingWithRightHand
  - wrist-Right : right_wrist du RainbowMan
  - shoulder_Right : right_shoulder du RainbowMan
  - tailleTab : 10
  - timerLimit : 0.1
  - timerCooldownLimit : 1
  - seuilDetectionRight : 3
  - seuilDetectionLeft : 3
  - seuilZoneMorte : 0.5

- SwippingUpWithBothHands
  - wrist-Right : right_wrist du RainbowMan
  - shoulder_Right : right_shoulder du RainbowMan
  - wrist-Left : left_wrist du RainbowMan
  - shoulder_Left : left_shoulder du RainbowMan
  - tailleTab : 20
  - timerLimit : 0.1
  - timerCooldownLimit : 2
  - seuilDetection : 2.5
  - seuil Zone Morte : 1

- Run and Punch
  - wrist-Right : right_wrist du RainbowMan

- hip-Right : right_hip du RainbowMan
- wrist-Left : left_wrist du RainbowMan
- hip-Left : left_hip du RainbowMan
- minTimeBetweenHands : 2
- maxTimeBetweenHands : 6
- tailleTab : 30
- tailleTabPunch : 10
- timerLimit : 0.1
- timerCooldownLimit : 1.0
- timerCooldownLimitRun : 2.0
- seuilDetection : 2.0
- seuilDetectionPunch : 2.0
- seuilZoneMorte : 0.5

› Military Pose
- head : head du RainbowMan
- hand-Right : right_hand du RainbowMan
- elbow_Right : right_elbow du RainbowMan
- shoulder_Right : right_shoulder du RainbowMan
- seuilDetection : 2.0
- EstSelectionnable (script optionnel à ajouter, permet d'appeler la fonction servant à revenir au menu principal)

Pour ajouter un nouveau geste, il faut simplement dupliquer un script existant et modifier le membre servant de référence, ainsi que les composantes qui seront stockées. Enfin, il faut changer la fonction de détection afin qu'elle définisse clairement après quel type de mouvement le geste voulu est détecté. Pour finalement intégrer un texte concernant ce nouveau mouvement dans le canvas, il suffit de dupliquer un de ces affichages de texte, et d'adapter les GameObjects associés aux variables du script pour lui indiquer quel booléen lui servira de référence pour savoir si le geste est détecté à un instant t.