Proposition de recherche

Détection de mouvement avec la Kinect

1 Le sujet

En 2010, Microsoft sortit la Kinect, un appareil connecté à la Xbox 360 intégrant la reconnaissance de mouvement en trois dimensions aux jeux vidéo. La Kinect permit une toute nouvelle approche à la façon dont on joue et ouvrit une multitude de possibilités pour les développeurs de l'industrie.

Le but de la recherche sera d'identifier les fonctionnalités et les capacités de la Kinect pour utilisation dans les jeux vidéo.

Le développement des capacités de la Kinect permettrait de ne plus avoir besoin d'utiliser de manette ou d'améliorer l'expérience avec manette déjà existante. On pourrait ainsi contrôler en partie ou en totalité l'action avec des gestuelles.

Présentement, la Kinect est principalement utilisée pour jouer à des jeux vidéo avec la Xbox 360 et la Xbox One. Elle peut aussi être utilisée avec un ordinateur sur la plateforme Windows et peut donc être utilisée dans n'importe quel domaine où la reconnaissance de mouvement est pertinente, comme la réalité augmentée, la santé ou la sécurité.

Nous savons comment utiliser la Kinect, ayant déjà joué avec des applications compatibles. La Kinect permet la détection de profondeur d'une image, la détection de couleur, la détection de mouvement, la reconnaissance vocale. Pour la capture de mouvement en général, nous prévoyons apprendre la théorie tout au long de la recherche. Nous savons ce que la Kinect peut faire, mais nous ignorons son fonctionnement.

2 L'enjeu

Le projet permettra d'acquérir plusieurs connaissances propres à la capture de mouvements. Les aspects explorés regroupent la détection d'un mouvement, de sa direction et de sa profondeur, en plus de la reconnaissance d'un motif en gestuelles. De plus, il sera nécessaire de maîtriser la détection d'un corps ou d'une partie spécifique d'un corps. Finalement, nous verrons l'intégration de ces concepts dans une application contrôlée par caméra.

La capture de mouvement est un concept de plus en plus présent dans la vie quotidienne, prouvant la pertinence de la recherche. La technologie évolue rapidement et offre de plus en plus de possibilités pour les développeurs et les utilisateurs. Aujourd'hui, la capture de mouvement fait partie de téléphones intelligents, de consoles de jeux vidéo, de systèmes de domotiques, etc. Cette technologie voit un progrès constant. La Kinect en est d'ailleurs à sa deuxième itération.

La recherche doit permettre une implémentation simple, rapide et efficace de la Kinect dans un moteur de jeu. Elle doit aussi faciliter l'exploitation des ressources offertes par la Kinect et par son kit de développement, en plus d'identifier le potentiel de la caméra (des fonctionnalités supplémentaires inconnues pour l'instant qui pourraient être implémentées dans le prototype).

La Kinect étant accessible au grand public, il sera possible de développer de nouvelles applications récréatives ou éducatives contrôlées par le mouvement, autant une caméra de surveillance pour poupon qu'un jeu vidéo. Il y a donc plusieurs applications futures possibles à coût raisonnable.

Cette recherche est donc souhaitée non seulement pour son utilisation en jeu vidéo, offrant une nouvelle expérience aux joueurs, mais aussi pour d'autres domaines tels que la santé, permettant d'offrir des services auparavant impossibles. Autant pour le design d'application avec capture de mouvements que pour l'exploitation même de la Kinect, la recherche ouvre des portes pour du développement futur.

3 Les objectifs

- Détecter une personne
- Détecter deux personnes différentes en même temps sans conflit
- Détecter les mouvements de la main
- Bouger la palette en fonction de la position de la main
- Jouer à deux sur la même caméra

Proposition de recherche

4 La méthodologie

Afin de suivre la progression et l'efficacité de la recherche, celle-ci sera menée par prototype. Tout au long de la recherche, nous aurons donc une application des résultats concrète et pratique.

Le prototype sera conçu avec Unity3D. Il s'agit d'un Pong qui devra intégrer les fonctionnalités de capture de mouvement de la Kinect. Dans cette version de Pong, le premier joueur qui comptera sept buts gagnera la partie. L'intérêt du prototype sera dans le contrôle des menus et des palettes de jeu, le tout étant contrôlé par mouvements de bras. Il pourra être joué par un seul ou deux joueurs, chacun contrôlant une palette avec sa main. La palette se déplacera selon la position de la main du joueur. La navigation dans les menus intégrera aussi la gestion de mouvements de type "swipe". Le projet nécessitera un exemplaire de la Kinect (pour 360, pour Windows ou pour Windows V2) en plus d'un PC avec le système d'exploitation Windows.

Le but du prototype sera d'expérimenter avec les types de contrôle par mouvement offerts par la Kinect. Il devra cibler autant la détection de position que la détection de mouvement et de profondeur. Il permettra donc d'exploiter l'essentiel des fonctionnalités offertes par la machine et de les intégrer à une application concrète.

5 Les incertitudes

À ce stade de la recherche, nous ne connaissons pas les méthodes de capture de mouvement. Nous ignorons comment capturer un mouvement naturel tout en permettant une marge d'erreur à l'utilisateur.

Nous avons aussi quelques doutes pour le multijoueur. Nous devrons nous informer sur la différenciation des mouvements de deux joueurs. Nous ignorons comment capturer simultanément les mouvements de deux joueurs. Il faut aussi conserver nos deux joueurs, même si ceux-ci se croisent ou entrent en conflit avec leur environnement. Par exemple, si un joueur quitte la caméra et revient, il doit conserver sa place de jeu.

6 Les limites

La Kinect offre la possibilité de faire de la reconnaissance vocale et la détection de couleurs, mais dû à la contrainte temporelle, ces aspects ne seront pas traités lors de cette recherche.

7 Les ressources

1. Matérielles
   - La Kinect pour XBOX 360
   - Un ordinateur
2. Logicielles
   - Le système d'exploitation Windows 7 ou 8
   - Unity3D 4.6.1
   - Le SDK de Kinect for Windows ver.1.8
   - Le packet de Kinect pour Unity3D ver.1.7
   - Le toolkit de Kinect pour développeurs Windows 1.8
3. Informationnelles
   - Documentation officielle sur le SDK de Kinect for Windows ver.1.8
   - Vidéo de la preuve de concept d'un Pong contrôlé avec la Kinect
   - Fichiers pour l'intégration du SDK de Kinect for Windows ver.1.7 dans Unity
   - Vidéo tutoriel de l'implémentation de la Kinect et de son SDK sur un poste de travail
   - Exemples d'intégration de la Kinect dans Unity
   - Exemple simple d'implémentation des mouvements de bras simples avec la Kinect dans une application
   - Tutoriel d'implémentation de gestuelles avec la main pour la Kinect dans une application

8 Échéancier

Périodes	Temps prévu	Descriptions	Responsables
26 janvier	3h	Recherche de documentation sur la Kinect et sur la capture de mouvements	PR et POB
28 janvier	3h	Création du prototype de Pong sans capture de mouvements	PR et POB

Proposition de recherche

2 février	4h	Programmation de la détection des mains	PR et POB
4 février	3h	Liaison des contrôles du jeu avec le positionnement de la main.	PR et POB
5 février	3h	Programmation d'un gestionnaire de gestuelles.	PR et POB
9 février	4h	Suite de la programmation d'un gestionnaire de gestuelles	PR et POB
11 février	3h	Ajout de gestuelles spécifiques au gestionnaire	PR et POB
12 février	3h	Liaison des gestuelles aux contrôles du jeu	PR et POB
16 février	4h	Publication de la recherche sur le site	PR et POB
18 février	3h	Préparation de la présentation orale	PR et POB
19 février	3h	Correction et ajustements à la documentation	PR et POB
23 février	3h	Présentation finale du projet	PR et POB

Devis technique de l'outil

Objectif de l'expérimentation

Le but de la recherche était d'identifier les fonctionnalités et les capacités de la Kinect pour son utilisation dans les jeux vidéo.

Nous avons donc ciblé les objectifs suivants en fonction du but donné :

- Détecter une personne
- Détecter deux personnes différentes en même temps sans conflit
- Détecter les mouvements de la main
- Bouger la palette en fonction de la position de la main
- Pouvoir détecter 2 personnes différentes en même temps
- Jouer à deux sur la même caméra
- Naviguer dans le menu avec la main
- Afficher la position de la main du joueur

Ces objectifs ont tous été remplis avec succès. Ils nous ont permis d'explorer les principes de reconnaissance de mouvements et de gestuelles.

Limites

Notre prototype garde la position des membres et articulations de notre corps pour contrôler un objet dans le jeu comme un curseur ou une palette. Il peut aussi reconnaitre des mouvements simples comme un balayement horizontal d'une main et un éloignement ou rapprochement vertical des deux mains. On peut aussi remettre les mouvements capturés sur un modèle et l'animer en temps réel.

Le prototype ne reconnait pas les mouvements plus complexes (tels qu'un rond, pousser ou tirer) et ne permet pas la détection de plusieurs mouvements en même temps. De plus, le balayement de la main est uniquement utilisé dans les menus et l'éloignement et rapprochement dans le jeu. Nous ne pouvons donc pas confirmer que les deux mouvements marcheraient bien conjointement. Finalement, le balayement vertical n'est pas supporté.

Éléments encore inconnus

Plusieurs éléments de la capture de mouvement avec la Kinect nous restent inconnus après les expérimentations. Les expérimentations n'ont pas permis d'exploiter les fonctionnalités

- La détection des doigts et de leurs gestuelles.
- La reconnaissance faciale.
- Toute fonctionnalité en dehors des limites du projet (reconnaissance vocale).
- L'attribution d'identifiants aux corps sur caméras. (La Kinect permet d'attribuer un ID de squelette pour conserver la priorité de détection. Par exemple, si on quitte la caméra et retourne dans le cadre, la Kinect nous reconnaît et nous garde en mémoire.
- Le near mode de la Kinect pour Windows n'a pas été exploité

Suivi des incertitudes

Une des premières grandes incertitudes que nous avions était le multijoueur, mais il fut très facile de l'implémenter. La Kinect le gère par elle-même et nous envoie toutes les données séparément.

Notre deuxième crainte était la capture de gestuelles qui fut relativement complexe. Nous nous sommes inspirés d'algorithmes déjà existants et des fonctionnalités offertes par le wrapper de Kinect pour Unity.

Nous voulions aussi assurer une marge d'erreur pour les gestuelles. Celle-ci fut un peu plus complexe à identifier afin de permettre un mouvement naturel, mais avec essai-erreur, nous avons trouvé ce que nous croyons être un juste milieu.

Référence

- Documentation officielle sur le SDK de Kinect for Windows ver.1.8

- Vidéo de la preuve de concept d'un Pong contrôlé avec la Kinect
- Fichiers pour l'intégration du SDK de Kinect for Windows ver.1.7 dans Unity
- Vidéo tutoriel de l'implémentation de la Kinect et de son SDK sur un poste de travail
- Exemples d'intégration de la Kinect dans Unity
- Exemple simple d'implémentation des mouvements de bras simples avec la Kinect dans une application
- Tutoriel d'implémentation de gestuelles avec la main pour la Kinect dans une application

Description de l'outil

Ce qui est nécéssaire

- Un ordinateur avec le système d'exploitation Windows 7 ou 8
- Un Kinect V1 pour Xbox 360 ou Windows
- Un câble USB compatible avec la Kinect V1 (non fourni avec certains modèles de la Kinect pour 360)

Fonctionnement

Pour le fonctionnement du prototype

1. Brancher la Kinect dans l'ordinateur
2. Attendre que les pilotes s'installent
3. Lancer l'exécutable du prototype

Une fois dans le jeu

1. Tendre la main vers la caméra et bouger le curseur sur le bouton « Jouer » assez longtemps pour effectuer un clic.
2. Choisir un mode de jeu grâce au curseur en le déplaçant sur le nombre de joueurs désiré.
3. Sélectionner un personnage en effectuant des balayements de la gauche ou de la droite. Un balayement vers la gauche affiche le prochain personnage disponible. Un balayement vers la droite affiche le précédent.
4. Si en mode deux joueurs, lever la main utilisée dans le jeu. Cette main sera utilisée pour contrôler la palette.
5. Lors d'une partie, bouger la main verticalement afin de déplacer une palette.
   - En mode 1 joueur, la main gauche et la main droite déplacent les deux palettes.
   - On peut éloigner ou rapprocher nos mains verticalement afin de grossir notre palette ou rétrécir la palette de l'opposant. En mode un joueur, cette fonctionnalité rétrécira toujours la palette de droite.

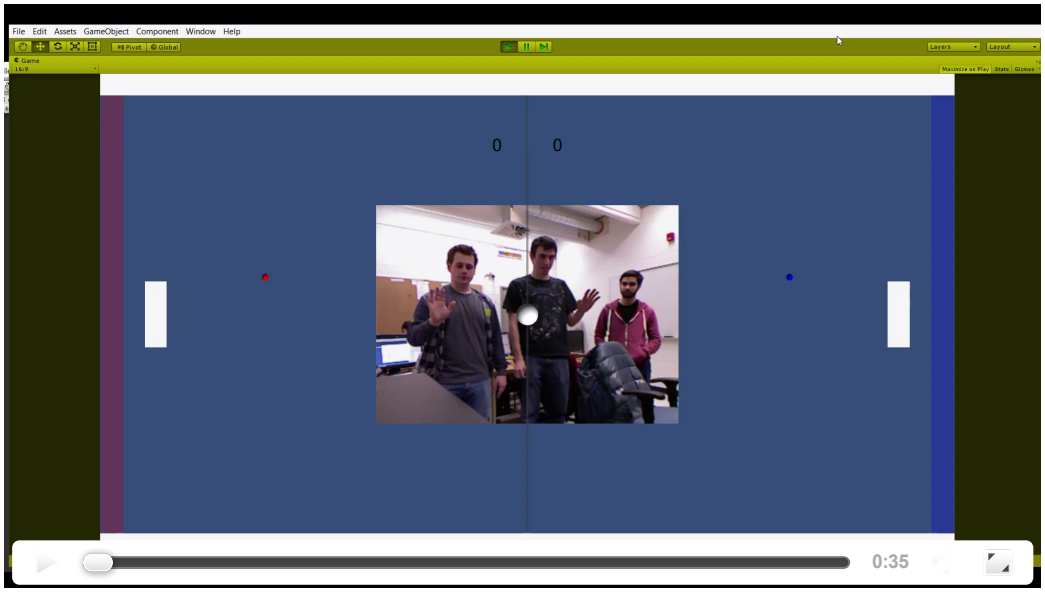
Paramétrage

Lors de la sélection de résolution d'écran, il faut choisir une résolution égale ou plus petite à la résolution de notre écran.

- Une fois dans le prototype, on peut lancer une partie à un ou deux joueurs.
- Il est possible de sélectionner un personnage dans les menus grâce au balayement horizontal.
- Chaque joueur peut choisir la main utilisée dans le jeu afin en la levant lors de l'écran de calibration (en mode deux joueurs).

Voir la section Fonctionnement pour plus de détails.

Billet du 26 janvier 2015



Résultats des recherches de la journée

- Récupération du code du jeu Pong
- Intégration de la Kinect dans Unity
- Intégration du contrôle par mouvement
- Gestion de deux joueurs simultanés

Récupération du code du jeu Pong

L'intérêt du prototype de Pong se trouvant dans l'intégration de la Kinect, nous avons décidé de récupérer le code d'un Pong déjà existant. Le code a été écrit par Philippe Proulx, professeur en jeu vidéo au Cégep de Ste-Foy et a été récupéré avec son consentement (Un gros merci pour son aide!). Bien sûr, le code nécessite plusieurs ajustements pour permettre l'intégration de la Kinect. Il y a donc beaucoup de modifications et de refactorisation à effectuer.

Intégration de la Kinect dans Unity

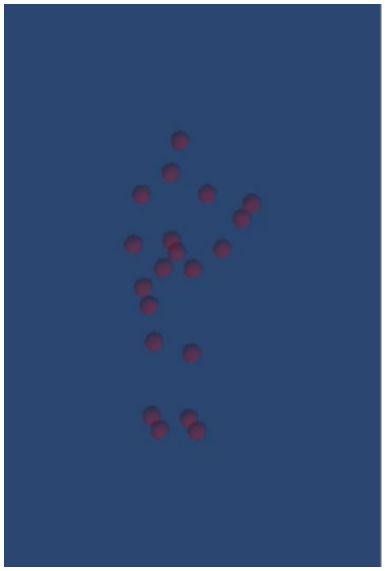
La première étape pour l'utilisation de la Kinect est son intégration dans l'environnement de développement de Unity. La caméra n'étant pas compatible par défaut, il a donc été nécessaire d'utiliser un package créé au préalable. Ce package a pour effet d'émuler le comportement de base de la Kinect offert par Microsoft. Le package est disponible ici. Il offre la gestion des articulations et des os dans un squelette détecté. On peut gérer jusqu'à deux squelettes en simultané et la détection de six corps.

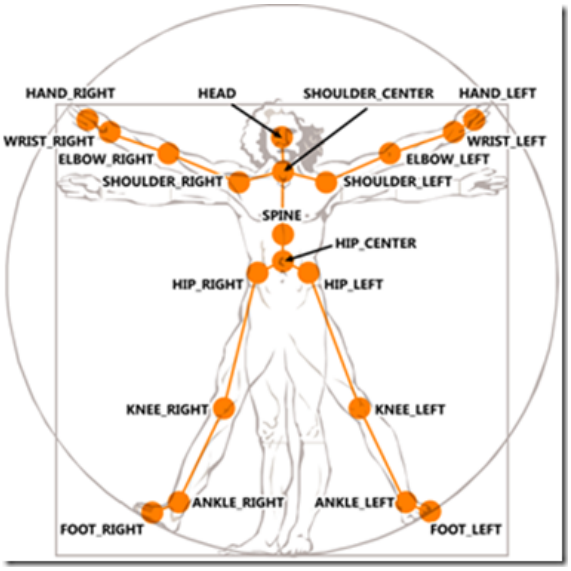
PAGES

Proposition de recherche
Devis technique de l'outil
Description de l'outil
Code de l'outil

BILLETS

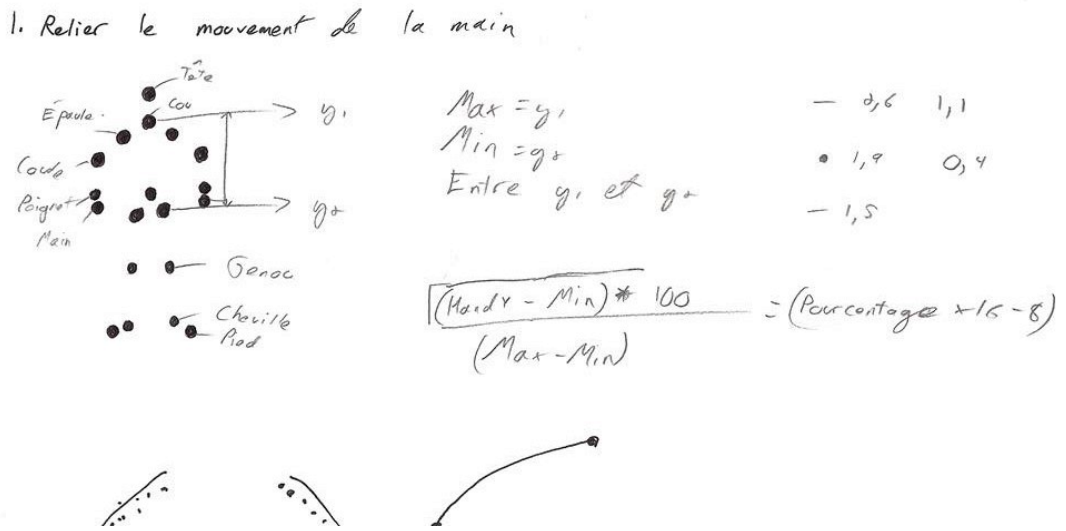
26 janvier 2015
28 janvier 2015
29 janvier 2015
1 février 2015
2 février 2015
4 février 2015
5 février 2015
9 février 2015
11 février 2015
12 février 2015





Intégration du contrôle par mouvement

Une fois la kinect fonctionnelle dans Unity, nous avons pu tester le déplacement d'une palette avec la main. L'algorithme prend en compte la distance entre l'épaule et les hanches pour créer un ratio avec la position de la main droite.



Ce ratio est alors réutilisé pour déterminer un point vertical dans le jeu. Par exemple, si notre main se situe à 60% en hauteur verticale entre notre épaule et nos hanches, la palette sera à une hauteur équivalant à 60% de l'écran de jeu.

Gestion de deux joueurs simultanés

La Kinect peut gérer les articulations de deux personnes. Nous pouvons techniquement contrôler le jeu avec n'importe quelle partie du corps de un ou deux joueurs. Par exemple:

- La main gauche du joueur 1 contre la main droite du joueur 2
- La main gauche du joueur 1 contre la main droite du joueur 1
- La genoux gauche du joueur 1 contre le coude gaughe du joueur 2
- ...

Voir le vidéo pour un exemple d'une partie à deux joueurs.

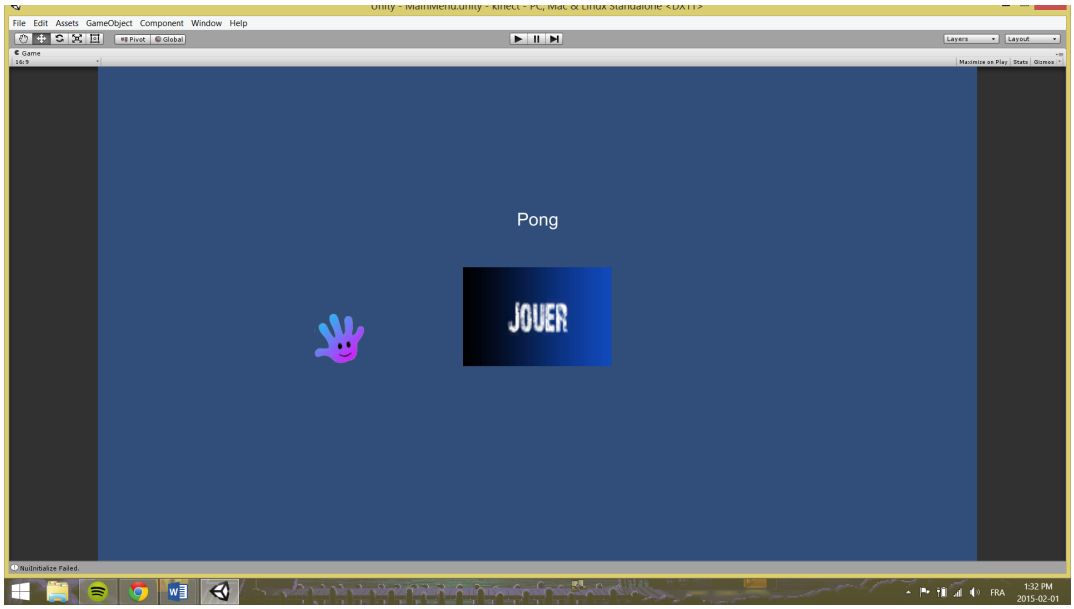
Billet du 28 janvier 2015

Résultats des recherches de la journée

- Intégration d'un menu fonctionnel
- Tests de gestuelles
- Utilisation de OpenNI et de NiTE

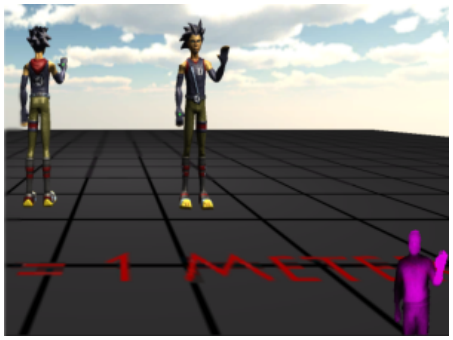
Intégration d'un menu fonctionnel

Nous avons ajouté un menu pouvant être contrôlé par les mouvements de la main. Le menu permet de presser un bouton en plaçant un curseur sur celui-ci. Lorsque l'on reste immobile assez longtemps sur le bouton, on le presse et passe à la partie.



Tests de gestuelles

Des premières expérimentations ont été effectuées dans Unity pour incorporer les gestuelles dans un projet. Des exemples sont d'ailleurs disponibles sur le « Asset Store ». Le projet principal exploré est celui de Rumen Filkov, un professeur autrichien de l'Université de Vorarlberg des Sciences Appliquées.



Lien vers le magasin

Nous avons donc exploré les possibilités de gestion des mouvements, tels que le balayement (ou « swipe ») dans quelconque direction, le saut, l'accroupissement, pousser et tirer. Une gestuelle importante est malheureusement manquante dans le projet, soit celle du « grip » (ou la fermeture et ouverture de la main).

Vidéo des expérimentations

Utilisation de OpenNI et de NiTE

OpenNI est un logiciel open source de type « Middleware ». Il permet l'intégration facilitée de la Kinect (et autres caméras intelligentes) dans un projet. Ici, il peut être utile avec la Kinect, permettant essentiellement de récupérer les fonctionnalités offertes par la caméra dans Unity. Malheureusement, puisque Unity utilise le framework 3.5 de .NET, il ne peut intégrer le SDK de la Kinect, celui-ci étant en 4.0. Il faut donc trouver un moyen de contourner cette limite, OpenNI semblant être une bonne alternative.

NiTE sert à faciliter la reconnaissance des mains, de la figure et de la profondeur et peut être utilisé conjointement avec OpenNI.

Nous explorons donc la possibilité d'utiliser ces « middlewares » dans notre projet, principalement pour la reconnaissance de la gestuelle de type « Grip ».

Exemple

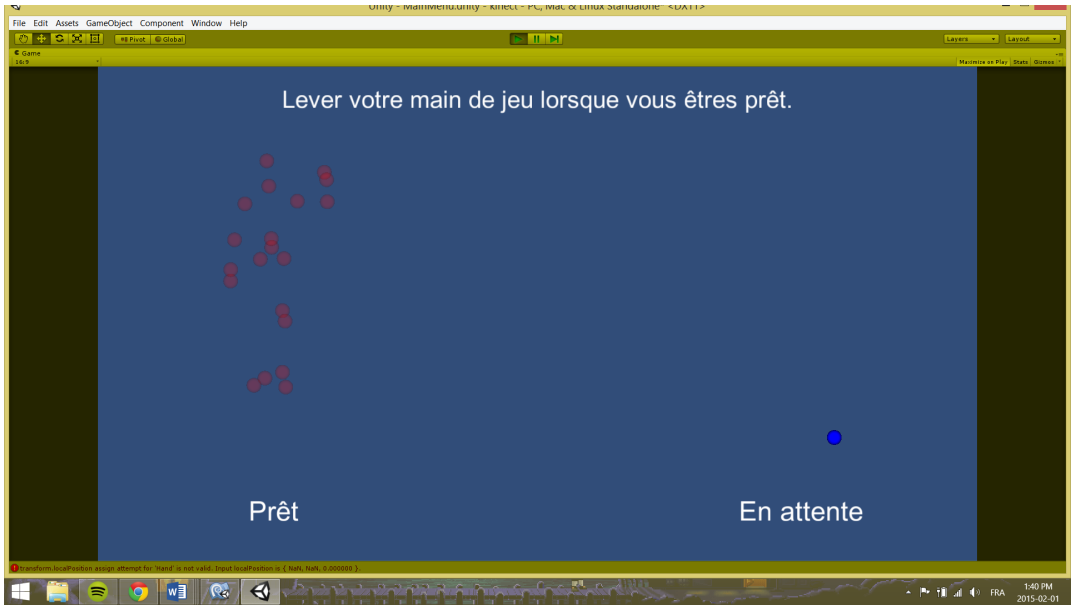
Billet du 29 février 2015

Résultats des recherches de la journée

- Sélection de la main dominante du joueur
- Utilisation d'un « wrapper » pour le SDK de Microsoft

Sélection de la main dominante du joueur

Chaque joueur peut maintenant sélectionner la main avec laquelle il jouera. On détecte premièrement les deux joueurs qui seront reliés à la partie et on leur permet ensuite de lever la main qui sera utilisée pour contrôler les palettes.



Utilisation d'un « wrapper » pour le SDK de Microsoft

Le « wrapper » actuellement utilisé pour intégrer le SDK de Microsoft dans Unity est assez complet, mais ne permet pas la détection de la forme de la main. Nous voulons pouvoir détecter si la main est fermée ou ouverte (gestuelle appelée « grip »), mais le paquet inclus ne le permet pas. Plusieurs possibilités s'offrent à nous :

- Recommencer le projet sur Visual Studio avec Win Form ou WPF
- Essayer d'intégrer OpenNI et NiTE
- Trouver un « wrapper » plus complet

Évidemment, nous ne pouvons pas recommencer le projet au complet. Nous sommes donc tombés sur un projet payant dans l'« Asset Store » de Unity incluant un exemple de gestuelle « agripper ». Il s'agit d'un autre projet du professeur Rumen Filkov, servant d'extra aux techniques explorées auparavant.

Le projet étant payant, nous avons envoyé un courriel à M.Filkov pour demander l'utilisation du projet gratuite dans le cadre d'apprentissage de la recherche. Nous pourrions ainsi ajouter des fonctionnalités à notre projet grâce au « wrapper » amélioré que M.Filkov offre.

PAGES

Proposition de recherche
Devis technique de l'outil
Description de l'outil
Code de l'outil

BILLETS

26 janvier 2015
28 janvier 2015
29 janvier 2015
1 février 2015
2 février 2015
4 février 2015
5 février 2015
9 février 2015
11 février 2015
12 février 2015

Billet du 1 février 2015

Résultats des recherches de la journée

- Obtention du code de M.Filkov pour agripper
- Tests du code

Obtention du code de M.Filkov pour agripper

Nous avons obtenu une réponse de la part de M.Filkov nous donnant un accès direct à son code. Il nous a autorisé l'utilisation dans le contexte d'éducation actuel, mais celui-ci ne peut être réutilisé pour un projet commercial ou repartagé à d'autres développeurs.

Comme promis, le code contient donc un « wrapper » pour Unity permettant l'utilisation de fonctionnalités du SDK de Microsoft. On peut donc accéder aux méthodes pour valider le geste « agripper » qui a été implémenté dans le SDK 1.7.

Il est important de noter que ce code permet l'utilisation de fonctionnalités offertes par Microsoft. Il s'agit uniquement de les implémenter dans Unity. Les exemples de gestuelles servent de références uniquement et devront être modifiés afin de répondre à nos besoins. Nous ne voulons pas uniquement réutiliser le code offert par le professeur.

Billet du 2 février 2015

Résultats des recherches de la journée

- Tentative d'intégration du « grip » dans le projet
- Début du code des gestuelles

Tentative d'intégration du « grip » dans le projet

Pierre-Olivier a tenté d'intégrer le code du grip dans le jeu Pong à l'aide du code obtenu lors 29 janvier dernier.

Les résultats sont peu concluants. Plusieurs approches ont été prises. Nous avons premièrement tenté d'utiliser tout simplement le « wrapper » de M. Rumen Filkov. Lorsqu'utilisé conjointement avec le « wrapper » d'origine utilisé pour notre jeu (voir le billet du 26 janvier 2015), celui-ci tente de réinitialiser la Kinect. En réalité, les deux « wrappers » doivent initialiser la Kinect afin de pouvoir être utilisés. La machine doit être initialisée par chacune des librairies, sinon, les autres méthodes ne seront pas fonctionnelles. Il y a donc automatiquement un conflit.

Billet du 4 février 2015

Résultats des recherches de la journée

- Mise en place d'un système de gestion des gestuelles
- Exploration des librairies
- Utilisation de l'émulateur de Kinect

Mise en place d'un système de gestion des gestuelles

Philippe a commencé l'implémentation d'un système de gestion de gestuelles. Celui-ci permet de reconnaître un geste par segments. Chaque geste commence par un segment de début (ex : La main droite sous la hanche). Ensuite, on retrouve des segments de milieu (ex : La main droite entre les hanches et la tête). Plus il y a de segments de milieu, plus la reconnaissance du geste est précise. Finalement, il y a un segment de fin qui conclue le mouvement (ex : La main droite au-dessus de la tête). Le mouvement test implémenté est le « clap » dans lequel on balaie verticalement les bras en les refermant sur eux-mêmes.

Le système devrait être réutilisable pour tout type de gestuelle.

Exploration des librairies

À l'aide de Depends (Dependency Walker), nous avons pu explorer le contenu des librairies Kinect utilisées. La librairie explorée ci-dessous est Kinect10.dll, soit la librairie officielle fournie par Microsoft. On retrouve ici les méthodes natives de la Kinect, soit le flux d'images, la reconnaissance audio, la reconnaissance de squelettes et d'autres contrôles de la machine elle-même. Ces méthodes sont utilisées par le « wrapper » du projet. Nous pouvons donc modifier ce « wrapper » pour accéder à d'autres méthodes qui nous semblent nécessaires.

0#	1 (0x0001)	N/A	N/A	0x000BB460
C	5 (0x0005)	7 (0x0007)	NuiGetSensorCount	0x00049AC0
C	6 (0x0006)	5 (0x0005)	NuiCreateSensorByIndex	0x00049B20
C	7 (0x0007)	0 (0x0000)	NuiCameraElevationGetAngle	0x00049A40
C	8 (0x0008)	1 (0x0001)	NuiCameraElevationSetAngle	0x00049A00
C	9 (0x0009)	8 (0x0008)	NuiImageGetColorPixelCoordinatesFromDepthPixel	0x00049770
C	10 (0x000A)	9 (0x0009)	NuiImageGetColorPixelCoordinatesFromDepthPixelAtResolution	0x00049800
C	11 (0x000B)	11 (0x000B)	NuiImageStreamGetNextFrame	0x00049670
C	12 (0x000C)	12 (0x000C)	NuiImageStreamOpen	0x00049550
C	13 (0x000D)	13 (0x000D)	NuiImageStreamReleaseFrame	0x00049700
C	14 (0x000E)	15 (0x000F)	NuiInitialize	0x000493F0
C	15 (0x000F)	17 (0x0011)	NuiSetFrameEndEvent	0x00049500
C	16 (0x0010)	18 (0x0012)	NuiShutdown	0x00049430
C	17 (0x0011)	20 (0x0014)	NuiSkeletonGetNextFrame	0x00049920
C	18 (0x0012)	22 (0x0016)	NuiSkeletonTrackingDisable	0x000498F0
C	19 (0x0013)	23 (0x0017)	NuiSkeletonTrackingEnable	0x000498A0
C	20 (0x0014)	24 (0x0018)	NuiTransformSmooth	0x000499B0
C	21 (0x0015)	4 (0x0004)	NuiCreateSensorById	0x00049C80
C	22 (0x0016)	16 (0x0010)	NuiSetDeviceStatusCallback	0x00049EE0
C	23 (0x0017)	14 (0x000E)	NuiImageStreamSetImageFrameFlags	0x000495D0
C	24 (0x0018)	10 (0x000A)	NuiImageStreamGetImageFrameFlags	0x00049620
C	25 (0x0019)	6 (0x0006)	NuiGetAudioSource	0x00049A80
C	26 (0x001A)	21 (0x0015)	NuiSkeletonSetTrackedSkeletons	0x00049970
C	27 (0x001B)	19 (0x0013)	NuiSkeletonCalculateBoneOrientations	0x00047DB0
C	28 (0x001C)	2 (0x0002)	NuiCreateCoordinateMapperFromParameters	0x00027AA0
C	29 (0x001D)	3 (0x0003)	NuiCreateDepthFilter	0x0002EB20

Il est aussi intéressant de comprendre le pourquoi et le comment du « wrapper ». Nous pouvons comprendre d'où viennent les fonctionnalités et en ajouter, puisque la librairie n'est pas actuellement utilisée en son entier.

Utilisation de l'émulateur de Kinect

Après l'utilisation d'une vraie Kinect pour les tests de notre prototype, nous nous sommes rendus à l'évidence qu'il est dérangeant d'utiliser la machine chaque fois que nous voulons tester une

fonctionnalité. Il est frustrant d'avoir à se lever à chaque lancement de l'application. Nous avons donc trouvé un émulateur de Kinect avec le « wrapper » utilisé dans le projet. Il permet d'enregistrer une série de mouvements et de la rejouer à tout moment dans l'application. Nous pouvons donc, par exemple, enregistrer une séquence dans laquelle nous naviguons dans les menus et lançons une partie. Cette fonctionnalité facilite grandement le développement et nous permettra d'avancer dans la création du prototype plus rapidement.

Billet du 5 février 2015

Résultats des recherches de la journée

- Implémentation du clap
- Début de l'implémentation du swipe
- Début de l'élaboration d'une machine à états pour les gestuelles

Implémentation du clap

Philippe a complété l'implémentation du mouvement de type clap dans le prototype. Celui-ci doit servir à réduire la grosseur de la palette de l'opposant ou agrandir notre propre palette. Le mouvement est reconnu avec succès. Étrangement, nous avons un certain problème avec l'émulateur Kinect. Le mouvement est reconnu avec la véritable Kinect, mais l'enregistrement fait à partir de l'émulateur n'est pas reconnu. De plus, le mouvement est reconnu dans notre scène Unity de tests, mais pas dans la scène de jeu.

Nous en sommes encore au début pour cette fonctionnalité.

Début de l'élaboration d'une machine à états pour les gestuelles

Afin de nous assurer que les gestuelles n'empiètent pas les unes sur les autres, nous devons créer une machine à état qui permet d'avoir une gestuelle à la fois et de calibrer la gestion de mouvements. Il n'y a pas encore d'élément concret pour la machine à états. Des discussions et élaborations préliminaires ont eu lieu.

PAGES

Proposition de recherche
Devis technique de l'outil
Description de l'outil
Code de l'outil

BILLETS

26 janvier 2015
28 janvier 2015
29 janvier 2015
1 février 2015
2 février 2015
4 février 2015
5 février 2015
9 février 2015
11 février 2015
12 février 2015

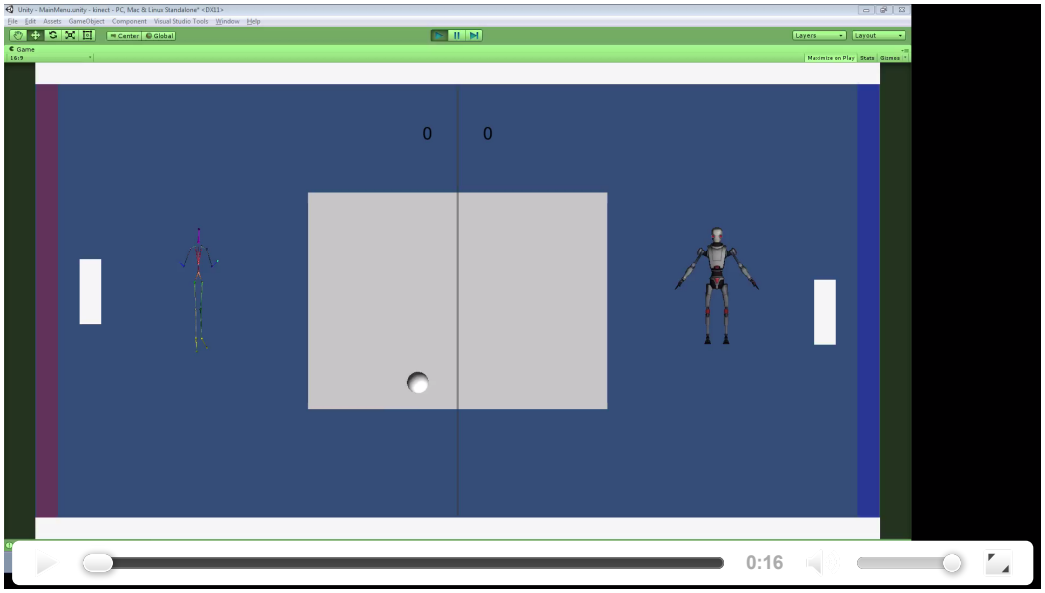
Billet du 09 février 2015

Résultats des recherches de la journée

- Fin de l'implémentation du clap

Implémentation du clap

L'implémentation du clap est terminée. Après beaucoup de test et débogage, nous avons trouvé pourquoi le clap était reconnu dans notre scène de test, mais pas dans le jeu. En fait, nous n'avions pas pris en compte que les donnés de la Kinect étaient mises dans un objet de la scène et donc, en fonction de l'orientation de notre scène, les axes de la Kinect avec le jeu pouvaient changer. Donc, dans notre scène test l'axe des x était l'axe des x de la Kinect, mais dans la scène de jeu c'était l'axe des z. Nous avons changé nos variables de position globale pour des variables de position locale.



Continuité de l'élaboration du balayement

Le balayement ou « swipe » progresse. Pierre-Olivier a ajouté la possibilité de reconnaître un balayement horizontal vers la gauche ou la droite. Le seul hic avec l'implémentation dans son état actuel est qu'il n'y a aucune gestion du mouvement de retour. Par exemple, si on balaie notre main vers la gauche, le retour de bras vers la droite recréé un nouveau balayement. Nous devrons donc ajouté une période où les mouvements après le balayement sont refusés tant que le bras ne retourne pas à son état normal (allongé sur le côté du corps).

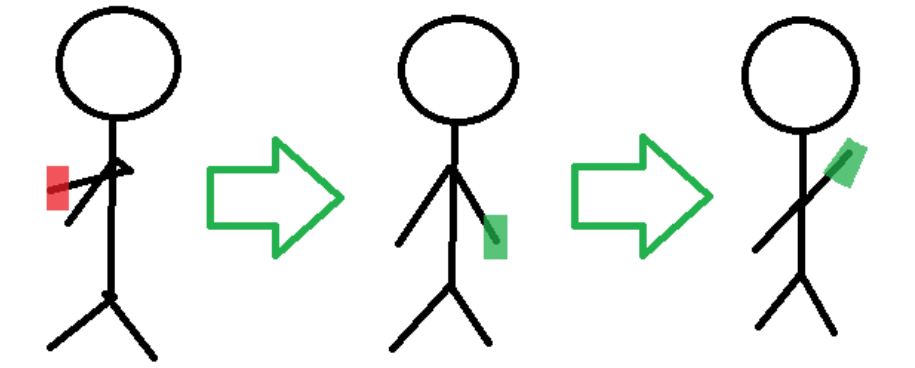
PAGES

Proposition de recherche
Devis technique de l'outil
Description de l'outil
Code de l'outil

BILLETS

26 janvier 2015
28 janvier 2015
29 janvier 2015
1 février 2015
2 février 2015
4 février 2015
5 février 2015
9 février 2015
11 février 2015
12 février 2015

Schéma simple d'un système de "Cooldown"



On finit le "swipe"

On retourne à
un état neutre

On peut refaire un
nouveau swipe

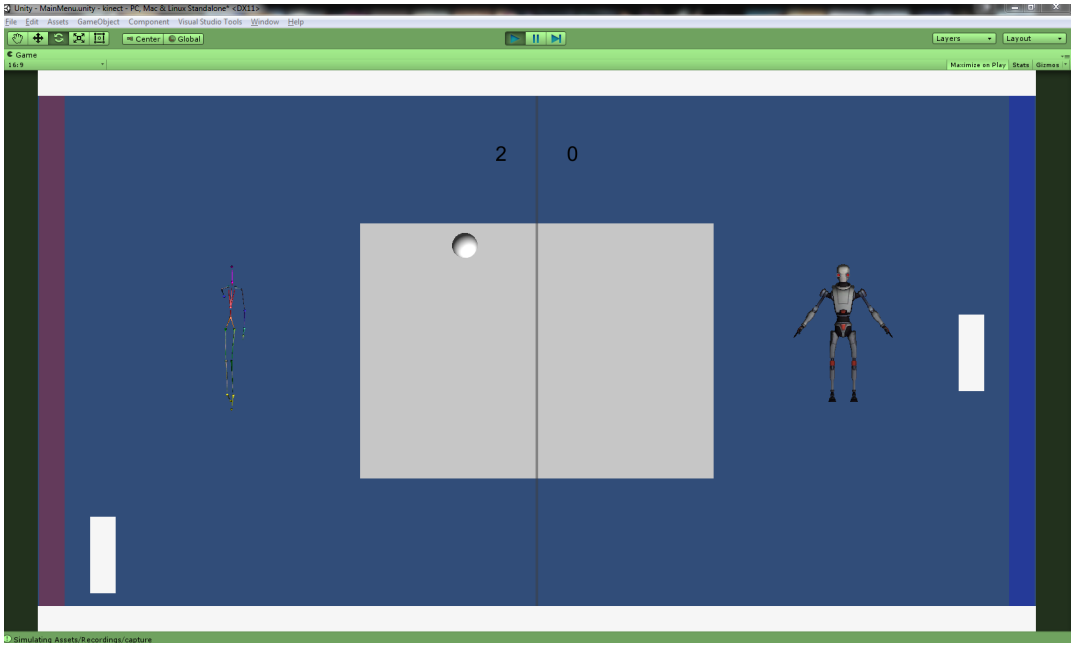
Billet du 11 février 2015

Résultats des recherches de la journée

- Animation d'un personnage 3D

Animation d'un personnage 3D

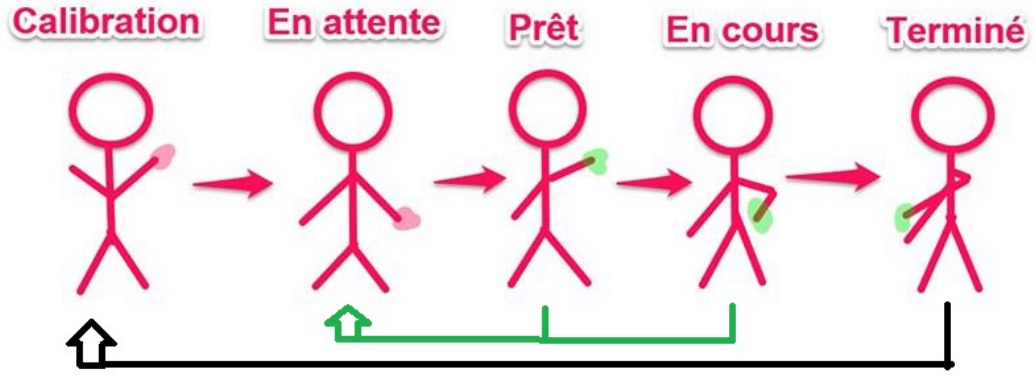
Philippe a téléchargé des modèles 3D dans le magasin de Unity et cherché une façon de pouvoir contrôler les modèles. Ce fut suffisant, puisqu'il faut simplement relier les parties du modèle avec les variables de la Kinect qui correspondent à la bonne partie du corps. Nous avons maintenant un personnage qui nous représente dans le jeu.



Ensuite Philippe a fait un menu pour choisir le modèle qui sera dans le jeu pour chaque joueur. Ce menu utilisera le « swipe » de Pierre-Olivier pour changer de modèle.

Fin de l'implémentation du balayement

Le balayement est fonctionnel avec une phase de « cooldown ». Voici un schéma de la fonctionnalité dans son état actuel.



On débute toujours par une calibration avec laquelle on attend que le joueur ait les deux bras au long de son corps. Par la suite, on vérifie s'il lève une main pour débuter un « swipe ». Le joueur doit ensuite faire passer rapidement sa main élevée entre ses hanches. S'il le fait trop lentement, il

PAGES

Proposition de recherche
Devis technique de l'outil
Description de l'outil
Code de l'outil

BILLETS

26 janvier 2015
28 janvier 2015
29 janvier 2015
1 février 2015
2 février 2015
4 février 2015
5 février 2015
9 février 2015
11 février 2015
12 février 2015

retourne en état d'attente, de même s'il abaisse la main.

Une fois le mouvement terminé, on retombe en calibration et on doit remettre nos bras au long de notre corps. Ainsi, lorsque le joueur ramène son bras à la suite d'un « swipe », on n'interprète pas le mouvement de retour comme un « swipe ».

Le balayement est donc prêt à être implémenté dans le menu.

Billet du 12 février 2015

Résultats des recherches de la journée

- Implémentation du swipe dans le menu de sélection de modèle
- Finition de l'outil

Pierre-Olivier a apporté quelques finitions au swipe puis nous l'avons ajouté dans le menu de sélection de modèles. Puis nous avons peaufiné les derniers détails de l'outil.

Nous avons fini notre outil!

PAGES

Proposition de recherche
Devis technique de l'outil
Description de l'outil
Code de l'outil

BILLETS

26 janvier 2015
28 janvier 2015
29 janvier 2015
1 février 2015
2 février 2015
4 février 2015
5 février 2015
9 février 2015
11 février 2015
12 février 2015

Code de l'outil

Procédure de compilation

Prérequis

- Unity3D 4.6.1
- Le SDK de Kinect for Windows ver.1.8
- L'outil

Ouvrir le projet avec Unity et compiler pour pc.