

## **Projet Kinexperience**

*Prose informelle de mi-parcours, Alex Andrix*

*Sur base des notes prises par l'équipe en semaines 1 et 2*

**Titre : Kinexperience**

**Date : Janvier – Février 2019**

**Image mise en avant**



### **Description courte**

L'écriture chorégraphique contemporaine est marquée par les études du théoricien de la danse Rudolf Laban. Dans l'étude de la Choreutique, il présente la kinésphère : un icosaèdre qui indique la limite naturelle de l'espace corporel de chacun où le mouvement a lieu.

Dans ce projet on propose d'explorer les mouvements de danse dans le cadre de cette spatialisation géométrique. Au-delà de l'intérêt technique et de recherche avec la grille de lecture et d'écriture de la danse proposée par Laban, l'oeuvre joue avec une panoplie d'effets graphiques déclenchés par les mouvements du corps.

## Etapes de réalisation

### Introduction

Suite à une première étape de réalisation effectuée en Juin 2018 (lien <http://labo-nrv.io/projet/kinexperience/>), l'équipe composée d'Ariane Cassimiro et d'Alex Andrix accueille en janvier 2019 l'artiste numérique **Yves Brozat** pour 4 semaines de résidence aux Subsistances. Le planning est confortable : 3 semaines d'expérimentations et de production au LaboNRV puis 1 semaine de mise en place au Plateau 2.

### Lancement

Ce deuxième temps de travail a été marqué par des réflexions quotidiennes sur la ligne artistique et le contexte du projet, qui ont orienté l'équipe sur les sujets d'exploration. Nous avons convenus d'aborder les 3 semaines de travail au LaboNRV en deux temps : **une phase d'ouverture** pour explorer un maximum d'opportunités techniques et artistiques, puis **une phase de cristallisation** pour construire une narration et une expérience.

Après une relecture des ouvertures envisagées en fin de sprint 1, nous avons commencé à identifier un certain nombre de chantiers techniques à explorer, de montées en compétences qui seraient nécessaires à la concrétisation de nos envies d'interaction.

- Intégrer dans Unity le **squelette** renvoyé par la Kinect
- Faire le bilan des techniques possibles pour la **représentation du corps** dans le projet
- Prendre en main le nouveau **Visual Effect Graph** arrivé en 2019 dans Unity
- Transfert de la version précédente du projet à la 3ème personne vers une vision en VR

Un brainstorming d'ouverture nous a permis de lister un certain nombre d'idées :

- Créer un objet (icosaedre) en réalité augmenté où viendrait danser une danseuse virtuelle
- **Corps fait uniquement de particules, continuité entre le corps et l'icosaèdre**
- Couleur et présence des particules en fonction de l'**énergie du membre déplacé**
- Créer un **hologramme** (pyramide) avec tablette
- Réflexions sur le lien entre monde virtuel et monde physique. Présence ou non d'objets physiques (icosaèdre taille réelle ?) comme **retour haptique**
- Faire des **interactions à partir des arêtes**
- **Interaction musicale en guide ou en réactif**
- Pointcloud VS squelette ? Possibilité d'utiliser **plusieurs Kinects** pour un pointcloud 3D
- Retour thermique et sensoriel de **flux d'airs** grâce à des ventilateurs/sèche-cheveux

### Phase d'auto-formation, rencontre d'obstacles et mise en place de solutions

La première semaine nous avons commencé à explorer les (nouveaux) outils que nous avons

identifiés auparavant. Nous vous proposons ci-dessous une séquence d'actions, de résultats ou de remarques que nous avons notés sur notre carnet de bord, **sans idée de forme**.

Travail pour faire fonctionner le HTC Vive du LaboNRV → la configuration de pièce ne semble jamais fonctionner (l'animation des bonhommes joués par le processus se bloque avant de pouvoir cliquer). Le problème venait simplement du fait que ni l'écran ni le Vive n'étaient reliés à la carte graphique spécialement équipée pour la VR !

Ajout d'une sortie jack sur le Vive → il est possible de démonter l'avant du casque pour y plugger toutes les sorties nécessaires, en outre pour une arrivée du son dans un casque audio, le pack prévoit un jack spécial qu'il suffit de brancher.

### *TUTO 1 - Créer un effet spécial avec Visual Effect Graph*

- Nouveau projet Unity en High Definition Render Pipeline
- Windows > Package Manager > Ajouter Visual Effect Graph et Post Processing
- Créer un GameObject Visual Effect
- Ajouter un fichier Visual Effect Graph (clic droit + Create dans l'arborescence)
- Lier les deux

Pour lancer le Game dans Unity en VR sur le casque : *File > build settings > player settings > XR settings > VR supported*

### *TUTO 2 - Animer une propriété grâce à une timeline (ex : mouvement de caméra pré-enregistré)*

- Créer un gameObject (empty) qu'on appelle timeline
- Créer dans Assets (clic droit) Create>Timeline
- Lier les deux (Timeline sur le GameObject)
- Afficher la fenêtre timeline (windows>Sequencing>timeline) et sélectionner le gameObject
- Sur cette fenêtre, cliquer sur Add puis Animation Track
- Créer un composant Animator sur le gameObject Camera et le lier à la timeline
- Cliquer sur le mode recording pour ajouter des keyFrames
- Double clique sur la sequence créer -> Cela ouvre une fenêtre animation où on peut éditer les valeurs des keyframes pour chaque paramètre, créer des courbes de transition entre les keyframes.

*SPECIAL TIPS* - LookAt : la caméra regarde toujours le centre du GameObject cible. LookAt.cs est un script C# qu'il faut affecter au GameObject de la caméra.

Inclusion de la Kinect dans un environnement Unity 2018.3 initié en HDRP, avec le package utilisé pour le sprint 1 du projet Unity, quelques débuts et ça fonctionne !

### *TUTO 3 - Intégrer la Kinect dans Unity (Readme package)*

- Juste lire le Readme (eh oui !) et ne pas inclure le package de manière classique. Le dossier KinectView n'est pas suffisant, il faut aussi les dossiers qui font le lien avec l'OS Windows comme le dossier *x86*.

Nous avons tenté de faire marcher l'ancienne version du projet (sprint 1 sous Unity 2018.1) pour se voir à l'intérieur de l'icosaèdre rouge et de tester les interactions déjà mises en place. Sans succès. Ni sur le poste du LaboNRV ni sur le pc personnel de Yves. Nous allons le re-crée rapidement dans tous les cas donc c'est sans trop de peine que nous avons renoncé.

### *TUTO 4 - Accéder à une variable de la Kinect dans un script Unity*

- 3 options pour partager des variables entre GameObjects : déclarer en public un GameObject dans l'autre et l'affecter, naviguer dans la relation parent/enfant, aller chercher le gameobject en lui assignant un tag
- Trouver les scripts intéressants qui donnent le squelette capté par la kinect :
  - Les copier-coller et les renommer dans les scripts de l'assets du projet
  - Créer 2 gameObject vides, **KinectController** et **KinectManager**, et leur affecter les scripts **BodySourceView.cs** et **BodySourceManager.cs** respectivement.
  - Créer un script **KinectBodyInteraction.cs** qui sera affecté à **KinectController** Celui-ci aura en entrée une variable public **BodySourceView** et récupèrera les données de position via des variables privées de type **Vector3**, dans la fonction **update** :

*Vector3 rH = bodySourceView.rightHandPosition;*

### *TUTO 5 - Lier squelette et particules*

Doc pour VEG -> <https://github.com/Unity-Technologies/ScriptableRenderPipeline/wiki/Visual-Effect-Graph>

Défrichage à la machette cérébrale du processus de **liaison entre VEG et éléments de scène** -> interaction entre la translation d'un cube et une variable de la turbulence VFX

Génération du squelette Kinect décortiquée -> des primitives cube et lignes sont générées en C# dans une méthode au lancement, mise à jour à chaque frame.

Il est donc envisageable de déporter cette génération scriptale vers une création de tout le corps -> trick magique du **Prefab** pour l'extraction ! (glisser-déposer le GameObject en play mode vers un dossier Prefab, pour le réimporter dans la scène hors play mode).

Dans notre cas précis reconstruire le corps hors play mode impliquerait cependant de réécrire une bonne partie du script originel du package Kinect Unity Pro. Nous avons finalement opté pour une mécanique de **Ghost**.

Cette technique que nous avons inventée - bon ok, d'autres ont probablement eu recours à cette même technique, mais on a trouvé un nom sympa et tout alors on est contents – consiste à positionner dans l'espace un certain nombre d'objets (des cubes par exemple) dont la position sera en lien direct avec les paramètres publics du VFX. Cette position peut être alors 'collée' à un membre du corps rendu par la Kinect, et le tour est joué ; l'information traverse les states logicielles et l'interaction est complète.

**Routine d'échauffement/réveil du corps avant la journée de travail**, visionnage de la vidéo [\*Major Lazer - Light It Up\*](#), idées de choix de représentations du corps.

Codage du script qui permet de **scale**, **translate** et **rotate** en harmonie, y compris les os qui lient les articulations du squelette -> on peut maintenant déplacer/tourner/redimensionner le squelette Kinect dans notre monde virtuel. En outre on peut soit le positionner loin du sujet pour un regard extérieur, soit faire en sorte qu'il épouse son propre corps. On teste, et ça marche très bien ! C'est un mélange entre deux technologies qui fonctionne. Pas besoin des trackers Vive, qui par ailleurs ne nous auraient donné que quelques positions sans idée de squelette global.

En l'occurrence on avait juste un souci avec la visualisation du corps dans la VR, et il nous a suffi de **re-calibrer le casque** par rapport à notre espace et placement du corps réel. Pour exemple ces valeurs prennent la forme de trois vecteurs (x, y, z) – de type Vector3, trois coordonnées pour chaque direction de l'espace - qui sont : Position (1,2 ; 7,9 ; -13,9), Rotation (0 ; 180 ; 0) et Scale (0,1 ; 0,1 ; 0,1). Ces valeurs nous laissent penser que les grandeurs sorties par la Kinect via le plugin Unity utilisé sont exprimées en mètres. Nous avons ajusté à la main la position du centre de l'icosaèdre par rapport au corps et la hauteur obtenue pour la translation est d'1m20, ce qui correspond à peu près à la hauteur du nombril (I.E. barycentre du corps) d'un homo sapiens sapiens moyen.

Recherche des modélisations du corps gratuites dans l'Asset Store de Unity > idée que **le sujet pourrait choisir entre plusieurs types de visualisations du corps dans un ensemble que l'on aurait prédéfini** (chercher à Skeleton - PBR - Animated - Low Poly, MCS Female).

Nous avons créé un effet visuel qu'on peut appeler *Boule Turbulente*, qui est en fait la combinaison de deux forces dans le block Update du VEG : **Turbulence** et **Conform to Sphere**. Cette boule, redimensionnée à taille humaine peut faire office de kinésphère conceptuelle. Nous avons donc essayé de rentrer à l'intérieur et d'en ressortir. Ce mouvement d'entrée dans cet espace privilégié de danse est très intéressant, et fort à vivre.

S'en suivent des essais en VR d'un effet de *regroupement d'énergie au creux de la main*, nous avons alors l'idée de dessiner la kinésphère avec nos propres membres. En outre garder la position initiale des particules et créer une force custom de retour vers cette position permettrait de visualiser la trace du mouvement, en plus de la kinésphère. Essais des effets sphère et turbulence tous les trois, prises de vidéos, upload des photos/vidéos sur Drive.

## TUTO 6 - Communiquer d'une tablette à Unity via l'app touchOSC

<http://thomasfredericks.github.io/UnityOSC/>

Pour les effets de traces, nous essayons une configuration où on laisse les particules à **vitesse nulle** sur une naissance en sphère avec rayon plus ou moins grand. Ça marche pas mal du tout. On raffine notre effet galactique avec une interaction simple : centrage sur la main droite et éclatement directement lié à la distance entre les mains. Petit ajustements pour que la zone d'interaction la plus riche se fasse quand les mains sont écartées à peu près à largeur d'épaules. En parallèle, on mène des essais de solutions pour **contrôle en remote OSC des différents paramètres Unity**.

On commence la création d'un système de forces qui permettrait de représenter l'icosaèdre en particules. Par une pose de forces sphériques sur chacun sommets, et un ajustement des rayons et intensités, on obtient une forme absolument surprenante : une sorte d'*hyper-icosaèdre*. C'est comme si on avait monté d'une dimension. Piste pour retrouver un icosaèdre : partir d'un tétraèdre et appliquer la même technique.

Enfin, un brainstorming sur la scénarisation aboutit à une **trame d'exploration progressive de son espace corporel**.

### Notes sur les 5 chantiers majeurs de recherche

- **Recherches 1** -> *liaison du squelette généré par la Kinect dans la scène avec les effets du Visual Effect Graph*
- **Recherches 2** -> *positionnement dimensionnement squelette avec soi, matérialisation de la Kinect dans l'espace virtuel*
- **Recherches 3** -> *explorer VEG à fond*
- **Recherches 4** -> *hacker la fonction Update de Visual Effect Graph*
- **Recherches 5** -> *Représentation du corps*

Le premier sujet a nécessité la mise en place d'une pipeline technique pour faire interagir les données de la Kinect avec les mécaniques de variables exposées par le nouveau système de VFX de Unity, notamment l'utilisation de ce que nous avons appelé des **Ghosts**, un ensemble de GameObjects qui font tampon entre l'univers d'interaction et l'objet qui porte la VFX.

Le second sujet nous a permis d'augmenter le script de base de lien entre Kinect et Unity d'une interface pour contrôler la position, la rotation et le redimensionnement global du squelette dans la scène, avec un peu de *calcul matriciel* et la lecture de la *doc Unity*, qui est soit dit en passant très bien construite.

Le troisième sujet a été réalisé à moitié, mais nous a permis de réaliser de beaux effets visuels à base de particules, et de les soumettre à la contrainte de proximité en portant le casque VR, ainsi que de

les tester en interaction. Sur ce point il en ressort globalement que **tout est plus lumineux dans le casque VR qu'à l'extérieur**, et que pour cette application-là il convient de réduire les effets de Post-Processing comme par exemple le Bloom qui a tendance à fortement éclairer la scène.

Nous n'avons pas eu l'occasion d'explorer le quatrième sujet, mais en lisant le code source nous avons pu estimer que ce ne serait pas trop difficile à mettre en place. Le seul inconvénient est qu'il faudrait réécrire le code source du package Visual Effect Graph de Unity, et que toute modification serait écrasée par les mises à jour suivantes. Avec un peu d'espoir un peut imaginer que les versions suivantes de VEG permettront l'écriture de shaders customisés.

Le sujet numéro 5 enfin, nous a amené à **imaginer les membres du sujet comme flux de particules**. En positionnant une source de particules au niveau de l'épaule et une force attractive radiale au niveau de la main, nous créons ainsi un bras fluide. Cette technique sera utilisée également pour matérialiser les arêtes de l'icosaèdre, par l'ajout de telles forces entre deux sommets adjacents.

# Quelques images





## **Mots clés**

kinésphère / numérique / danse / corps / motion capture

## **Crédits / Collaborateurs**

LaboNRV

Les Subsistances

Theoriz Studio (sprint 1)

## **Auteurs**

Artiste chorégraphe : Ariane Cassimiro

Artiste numérique : Yves Brozat

Artiste numérique : Alex Andrix (semaines 1 et 2)

## **Prose de compte-rendu à mi-parcours**

Alex Andrix

## **Matériaux / machines / logiciels employés / technologies**

3D creation software -> *Unity (version 2018.3.0f2)*

Body motion detection -> *Kinect v2*

Particle systems -> *Unity HDRP Visual Effect Graph*

VR headset -> *HTC Vive*

## **Fichiers et liens associés**

<https://www.les-subs.com/artiste-en-residence/ariane-cassimiro-et-alex-andrix/>

<http://labo-nrv.io/actu/ariane-cassimiro/>

<http://labo-nrv.io/actu/kinexperience/>

<http://labo-nrv.io/projet/kinexperience/>

<http://alexandrix.com/kinexperience>