

Programmation

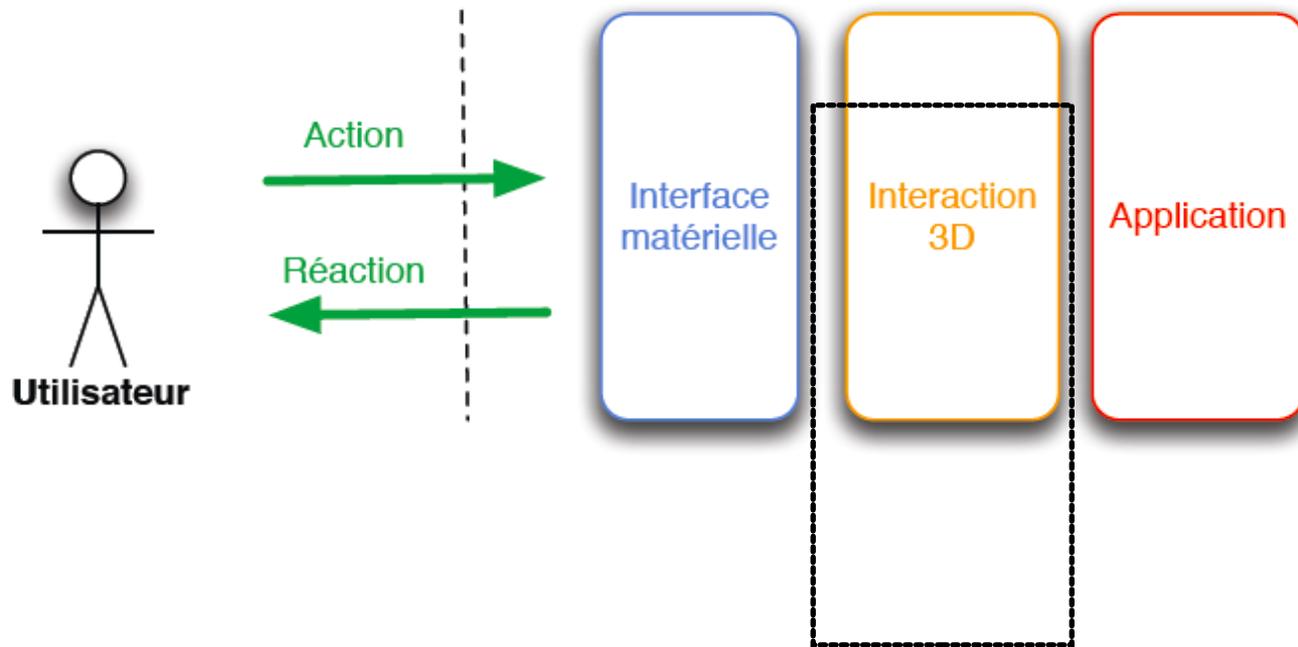
De techniques d'I3D

Frédéric DAVESNE (IGR UEVE)

frederic.davesne@ibisc.univ-evry.fr

Architecture générale d'un système d'I3D classique

[Otmane, 2010]



- 3 couches logicielles interconnectées
- Possiblement plusieurs PCs interconnectés

Formalisme générique d'une méthode d'I3D



IR: Informations reçues du monde Réel

EV: Etat du monde Virtuel

AR: Action sur le monde Réel

AV: Action sur le monde Virtuel

Séparation en 3 couches différentes

- Couche dédiée à la récupération des données formant le vecteur IR
 - pour chacune des techniques d'I3D utilisées
 - Mise à jour à chaque pas de temps
- Couche dédiée à l'implémentation des techniques d'I3D utilisées pour l'application
 - variables associées aux différents objets virtuels
 - variables générales associées à l'environnement virtuel: vecteur EV
- Couche dédiée à l'application
 - variables associées à l'application
- Communication entre les couches?

Plateforme de RV/RA EVR@

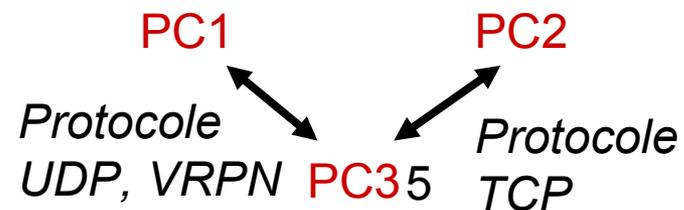
<http://evra.ibisc.univ-evry.fr>



3 PCs serveurs

- PC1 - Tracking ART
- PC2 - SPIDAR
- PC3 - Graphique, Applications

Communication



Virtual Reality Peripheral Network VRPN

- Librairie Open Source C++, site <http://www.cs.unc.edu/Research/vrpn/index.html>
- Gère un ensemble de périphériques dont
 - Kinect
 - Wiimote
 - ARTTracking
- Les périphériques sont regroupés en 3 types de périphériques:
 - Tracker
 - Analog
 - Button

Le type Tracker contient une position 3D et une orientation (quaternion).

Le type Analog est utilisé pour les périphériques tels que joystick ou souris

Le type Button est associé à une réponse binaire : bouton de souris, de joystick, etc.

Virtual Reality Peripheral Network VRPN

Exemple

-Souris

- Associée à deux canaux de type Analog (positions en X et Y) et 3 canaux de type Button

-Flystick (ARTTrack)

- Associé à un canal de type Tracker (position 3D + orientation 4D) et 7 canaux de type Button

-Kinect

- Associé à 24 canaux de type Tracker (position 3D + orientation 4D de 24 parties du corps)

Virtual Reality Peripheral Network VRPN

- Système Client/Serveur
 - Protocole réseau UDP
 - Adapté lorsque l'application de RV est distribuée sur plusieurs PC dédiés chacun à un périphérique

Virtual Reality Peripheral Network VRPN

-Coté serveur

Exemple de codage de la gestion d'un périphérique de type Tracker

```
void
myTracker::mainloop()
{
    vrpn_gettimeofday(&_timestamp, NULL);

    vrpn_Tracker::timestamp = _timestamp;

    // Les données sont récupérées du périphérique connecté au PC serveur VRPN
    static float angle = 0; angle += 0.001f;
    // le vecteur pos contient la position 3d du tracker
    // Ici on met juste des valeurs arbitraires
    pos[0] = sinf( angle );
    pos[1] = 0.0f;
    pos[2] = 0.0f;
```

Virtual Reality Peripheral Network VRPN

-Coté serveur

Exemple de codage de la gestion d'un périphérique de tracking

```
// Le tableau d_quat array contient les valeurs de l'orientation du tracker, mises dans un quaternion
```

```
    // On met ici des valeurs arbitraires
```

```
    d_quat[0] = 0.0f;
```

```
    d_quat[1] = 0.0f;
```

```
    d_quat[2] = 0.0f;
```

```
    d_quat[3] = 1.0f;
```

```
    char msgbuf[1000];
```

```
    d_sensor = 0;
```

```
// codage des valeurs du capteur dans le buffer du serveur
```

```
int len = vrpn_Tracker::encode_to(msgbuf);
```

```
    server_mainloop();
```

```
}
```

Virtual Reality Peripheral Network VRPN

-Coté client (récupération des données d'une souris)

```
#include "vrpn_Analog.h"
```

```
#include <iostream>  
using namespace std;
```

```
void VRPN_CALLBACK handle_analog( void* userData, const vrpn_ANALOGCB  
a )  
{  
int nbChannels = a.num_channel;  
for( int i=0; i < a.num_channel; i++ )  
{  
cout << a.channel[i] << " ";  
}  
cout << endl;  
}
```

Virtual Reality Peripheral Network VRPN

-Coté client (récupération des données d'une souris)

```
int main(int argc, char* argv[])
{
    vrpn_Analog_Remote* vrpnAnalog = new
    vrpn_Analog_Remote("Mouse0@localhost");
    vrpnAnalog->register_change_handler( 0, handle_analog );
    while(1)
    {
        vrpnAnalog->mainloop();
    }
    return 0;
}
```

Codage des objets en I3D

3 caractéristiques différentes coexistent

- Graphique/Physique (fait par un modéliser type 3DS Max, Maya, Blender)
 - Intersection possible
 - Masse
 - Cinématique
- EV lié à l'environnement virtuel
 - Etat de l'objet
 - Libre, sélectionnable, sélectionné
 - Position, orientation
 - Texture, couleur
- Hiérarchie dans le graphe de scène

Sélection des objets en I3D

- Plusieurs approches possibles
 - Selon la métaphore d'I3D utilisée
 - Selon le critère lié à l'état *sélectionnable*
- Détection de l'intersection de l'avatar avec l'objet graphique
 - Colliders associés à l'objet graphique

OU

- Proximité de l'avatar avec la position de l'objet
 - Définition d'un seuil <-> état sélectionnable