

# Lezione 9 10/04/2024

## Interactions in virtual worlds

Le interazioni nel virtual world: system control si interagisce con il sistema per fare qualcosa fuori all'environment virtuale.

Una volta selezionato un elemento si va a manipolarlo, e poi c'è la navigation, ovvero come l'utente si muove e controlla il virtual environment.

## Fundamentals of Human–Computer Interaction in VR/AR

La human-computer interaction è la ricerca che va ad analizzare come l'essere umano e il software interagiscono, studia il design dell'applicazione, la valutazione di come gli utenti interagiscono con l'applicazione, anche sui nuovi approcci uomo-macchina.

Tra gli aspetti di usabilità abbiamo l'utilità dell'applicazione, quanto è efficiente (in relazione all'obiettivo, quindi la fatica dell'utente per fare il task), quanto è effettiva (cioè quanto del goal riesce a fare), la learnability (quanto è difficile imparare ad usare l'applicazione), la soddisfazione soggettiva e l'user experience (il lato estetico dell'app, la piacevolezza, il flusso logico).

Per la parte di VR-AR non si può utilizzare tutto quello che è diventato standard nei pc (WIMP).

La parte di interazione naturale 3D include tutto quello che va a sfruttare interazioni naturali, come quando l'utente interagisce con il mondo, per esempio muovendo la mano.

Ci sono altri tipi di interazioni che non hanno un riferimento con una naturale, vengono dette interazioni magiche e permettono più possibilità e più funzionalità.

Le interfacce supportano la manipolazione diretta se l'utente può modificare la rappresentazione di un oggetto virtuale tramite un input device, ricevendo immediatamente un feedback. In questo modo si può trattare gli oggetti reali e quelli virtuali nello stesso modo, manipolarli nello stesso modo.

## System control

Alcune tecniche delle interfacce 2D possono essere usate nel mondo virtuale in modo limitato, dato che menu, bottoni e toolbar non sono realistici, e quindi vanno in conflitto con la "willful suspension of disbelief" perché questi comandi non hanno un equivalente nel mondo reale, oppure l'implementazione non è praticabile.

I concetti classici del system control sono i menu, i 3D widget, tangibles, voice commands e gestures.

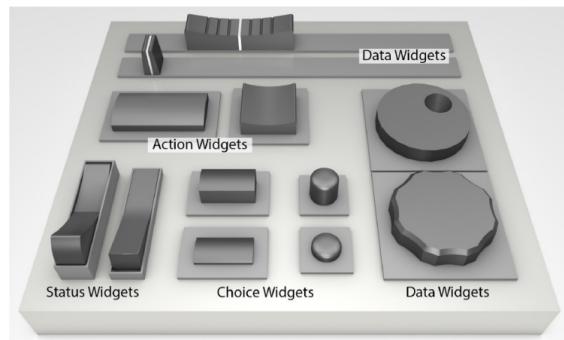
### Menu

I menu vengono presi dal mondo 2D e possono anche essere mantenuti 2D, in modo piatto. Ci sono approcci diversi, alcune volte viene fatto comparire con un bottone sul controller, oppure può essere lasciato fisso su una superficie nel mondo.

Il lato positivo è che tutti sanno come funzionano.

### 3D widgets

Sono oggetti che nella forma richiamano oggetti del mondo reale: manopole, bottoni, ... servono a richiamare una specifica funzionalità, quindi sono più immediati perché si basano sulla conoscenza pregressa dell'utente che sa come funzionano nel mondo reale.



### Tangibles

Anche chiamati props. Sono oggetti fisici, reali. Se l'utente li tocca, li usa come tools nel mondo virtuale. Quindi il sistema riconosce l'oggetto fisico che è inserito nel mondo e ne associa le funzionalità.

Toccando con mano l'oggetto fisico, questa diventa un'azione molto naturale. Il sistema deve quindi dare un feedback immediato.



## Voice commands

Di buono non coprono nessuna parte del mondo virtuale (a differenza del menu), però vanno imparati, bisogna sapere come interagire con il sistema, poi ci sono problemi di riconoscimento vocale (anche per colpa di suoni presenti nell'ambiente).

## Gestures

Associo a particolari movimenti della mano e delle dita dei comandi. Questi sono ancora più difficili dei voice commands perché vanno per forza imparati, a meno che vanno a sfruttare la conoscenza pregressa dell'utente, che non è detto che tutti gli utenti abbiano (come le due dita per fare lo zoom, o per scorrere una pagina)

## Selection

La selezione è l'operazione di individuare un oggetto specifico, o una parte del virtual world, con cui interagire.

Questo è problematico perché è difficile capire quale utente sta indicando in 3D perché c'è tutta la parte di profondità.

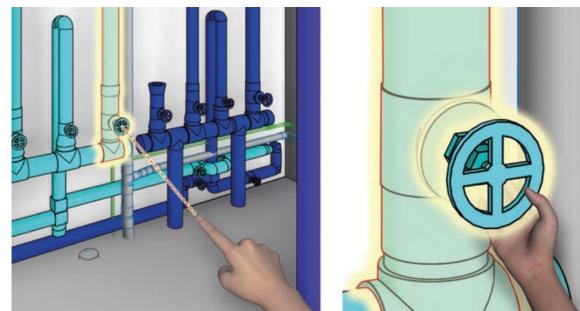
C'è anche il problema dell'occlusione. Anche se l'oggetto è lontano, se il puntatore utilizzato non è preciso diventa difficile.

## Pointing

Uno degli approcci di base è quello basato sul pointing. L'utente è in grado di mirare all'oggetto con cui interagire, di solito si utilizza l'indice della mano dell'utente, anche mostrando un raggio virtuale che aiuta la selezione.

Alternativamente si possono utilizzare altri device come stick.

Ci sono due approcci: il **color picking**: si crea la scena, ogni oggetto viene colorato diversamente, si prende il vettore che dice dove sta puntando l'utente che andrà ad incrociare un pixel della scena, prendo il colore di quel pixel e in quel modo deduco l'oggetto a cui sta puntando l'utente. Altrimenti si usa il **ray-casting**, dove si mostra un raggio che è un rigid body, e in base alle collisioni che percepisce ci dice qual è l'oggetto selezionato.



Il **pointing device** può essere **diretto** (pointing stick) se può essere usato per posizionare un cursore 3D direttamente. Quindi è in grado di definire le coordinate assolute, l'operazione è semplice ma può essere stancante e inaccurata dopo tempo, e l'utente potrebbe coprire delle parti del mondo per esempio con un braccio, nascondendole.

Altrimenti gli **indirect pointing devices** (mouse) possono ridurre questi svantaggi, individuando l'oggetto selezionato in modo indiretto, spostando il cursore indirettamente. Questi richiedono una fase di familiarizzazione per la coordinazione occhio-mano.

Le **local interactions** permettono all'utente di orientarsi velocemente grazie alle loro esperienze pregresse.

Le **remote interactions** sono interazioni che non esistono nel mondo reale, ma che permettono una serie di attività senza spostarsi (per esempio per raggiungere un elemento lontano). La **Fitt's Law** dice che più un oggetto è lontano da noi, e più tendenzialmente sarà difficile selezionarlo, richiederà più tempo.

C'è quindi un range ampio di device tra cui scegliere, può succedere che l'utente non capisca come deve interagire se si ritrova diversi metodi di interazione in momenti diversi. Questi sono chiamati **mode errors**, non sono errori di programmazione ma sono errori di progettazione nell'interazione uomo macchina. Può anche capitare che l'interazione non risponda come l'utente si aspetta, e questo può disorientarlo.

Il **ray-casting** è il concetto dove si utilizza un raggio, che parte magari dal dito o dall'occhio dell'utente, e in base a cosa incrocia, l'elemento viene selezionato. Questa è considerato il metodo più efficace al momento. Ci sono quindi i problemi dati dalla distanza dell'oggetto.

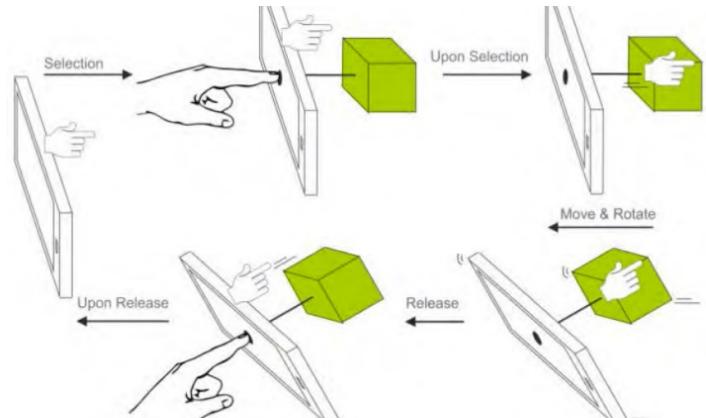
Il ray casting ha i problemi di occlusion, di selezioni accidentali per esempio se un oggetto si interseca con un altro.

C'è anche la tecnica "flashlight" dove viene proiettato un cono.

**La go-go technique** permette l'estensione infinita di un braccio che virtualmente si allunga e raggiunge l'oggetto. Questo non è realistico ma può essere comodo, perché non è necessario che l'utente si sposti per essere preciso.

### **Homer technique (spinge verso l'oggetto):**

- "Hand-centered Object Manipulation Extending Ray-casting"
- A beam is also extrapolated from the current hand position
- If the ray hits an object, the object is not manipulated at the endpoint of the ray, but the virtual hand is moved to the position of the object
  - This eliminates the dependence on angular accuracy and allows fine selections and manipulations of the target object



**L'image layer technique** è come se fosse un'immagine su cui l'utente può interagire, eliminando quindi il problema di interagire in 3D, ma lavorando in 2D.

Nella tecnica del **mondo in miniatura** viene mostrata una miniatura del mondo in cui l'utente si trova, che essendo piccolo l'utente ha alla propria portata qualunque elemento che può quindi selezionare.

Questo non è realistico, è una tecnica esocentrica.



## **Manipolazione**

Dopo aver selezionato un oggetto, vogliamo poterlo manipolare. La **manipolazione** è il cambiamento interattivo dei parametri che caratterizzano l'oggetto (posizione, orientamento, forma, peso, velocità, apparenza).

Le tecniche di selezione e manipolazione dovrebbero quindi essere le stesse, non diverse. Chiaramente sarebbe più naturale usare le mani per un'esperienza migliore rispetto ad un controller per esempio.

Anche qui ci sono le tecniche di manipolazione **realistiche**, e quelle **magiche**.

Poi ci sono quelle **esocentriche**, dove l'utente è nell'ambiente, ha una percezione di quello che è vicino a lui, ha una prospettiva, e una modifica dell'oggetto realistica, immersiva. Il problema è che l'oggetto deve essere per forza in prossimità dell'utente.

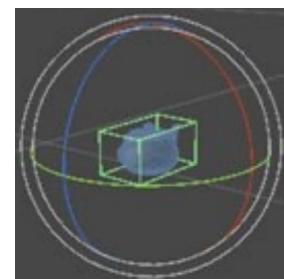
Con la **manipolazione egocentrica** è come se l'utente facesse un passo indietro, avendo una visione globale dell'ambiente, andando ad utilizzare un'interazione diversa, non diretta, per manipolare l'oggetto.

## Arcball

Invece che usare il controller per manipolare l'oggetto, facciamo comparire gli archi intorno all'oggetto, con cui l'utente può interagire per manipolarlo.

Una volta selezionato l'arco, l'interazione è sostanzialmente 2D.

Questo serve a modificare l'orientamento dell'oggetto.



## Virtual hands

L'utente interagisce con l'ambiente in maniera realistica, come farebbe nel mondo reale. L'utente può manipolare oggetti che possono raggiungere, che sono nella vicinanza. Le interfacce di questo tipo sono chiamate **natural user interfaces**.

D'altro canto le **super natural user interfaces** sono quelle che sono ispirate alla fantascienza, alla magia (es teletrasporto).

## Pointing gestures

Prima le utilizzavamo per il system control, qui le usiamo per la modifica. Dopo aver selezionato l'oggetto, utilizzo delle gesture per modificarlo. Con gesture si intende per esempio il gesto di apertura, e l'oggetto diventa più grande. Quindi

non ho l'oggetto virtuale nella mano virtuale. Anche qui l'utente deve conoscere le gesture.

## Navigation

Come fa l'utente a muoversi nel virtual world? Come nel mondo reale, è divisa in due parti: il **wayfinding** e il **traveling**.

### Wayfinding

Il wayfinding è la parte dove bisogna capire dove ci troviamo e dove vogliamo andare. Ci sono varie componenti: **analysis, planning e decision**.

Mi devo orientare nel virtual world, devo fare una mappa cognitiva di dove sono, per capire dove devo andare. Questa parte viene fatta in automatico da ognuno di noi, in modi diversi. Bisogna quindi tenere a mente come si può aiutare l'utente a svolgere queste attività cognitive in maniera veloce e semplice.

Il primo step è la creazione della **landmark knowledge**, l'utente cerca dei punti di riferimento nello spazio. Quindi si può aiutare l'utente inserendo dei landmarks chiari nel virtual world.

Lo step successivo è la **route knowledge**, è il procedimento nel cui l'utente si forma nella testa il percorso da dov'è a dove deve andare. Quindi se non ha fatto lo step di capire dove si trova nel mondo, non lo può fare. Si può aiutare l'utente con una bussola virtuale, o altro.

L'**overview knowledge** è la visione completa, non solo di quello che si ha intorno, ma di tutto nel complesso. Questa è complessa, richiede più tempo perché richiede la navigazione di tutto il virtual world. La landmark e route knowledge messe insieme pian piano formano l'overview knowledge.

### Traveling

Qui si esce dalla parte cognitiva e passiamo a come muoversi. Ci sono 3 diversi task associati al traveling: **exploration, search e maneuvering**.

**Exploration** è quando si naviga il virtual world, senza uno scopo preciso. Questo tipo di task è fatto per orientarsi.

Il task più preciso invece è il **search**, è goal oriented, devo andare in un punto preciso e quindi mi muovo per raggiungerlo.

**Il maneuvering** è il movimento piccolo dell'utente, nel suo intorno, cioè se devo andare da qualche parte per esempio mi guardo intorno, mi muovo poco, e poi parto.

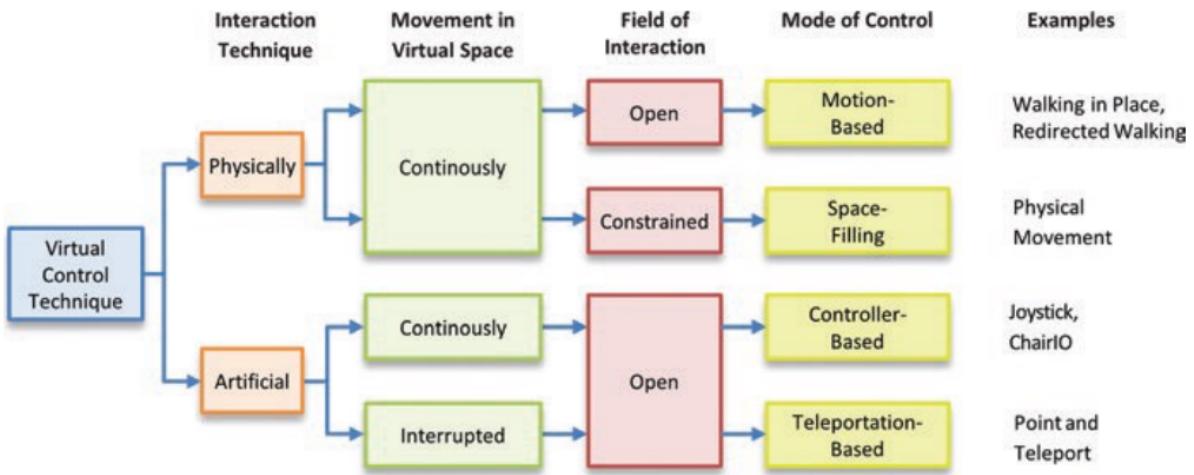
Che tecniche abbiamo per fare il traveling? Nei sistemi di augmented reality si cammina, invece in VR questo non è sempre fattibile.

Si possono avere tecniche di traveling di tipo **implicito** o **esplicito**. Molti sistemi usano la **virtual reality locomotion**, il concetto è che sposto la virtual camera in qualche maniera, creando un vettore per lo spostamento, potrei usare un controller, uno stick, puntando dove voglio andare. Però se l'utente ha in mano uno stick per muoversi, avrà solo una mano per fare tutte le altre operazioni di selezione/manipolazione.

Altri approcci usano lo sguardo, creando un **gaze vector**. Può essere usata la posizione dell'headset o quella dello sguardo, usando la direzione per far spostare l'utente. Questo è difficile perché si porta indietro la complessità di fare tracking dello sguardo, e poi se l'utente deve usare lo sguardo per decidere dove andare, non può guardarsi in giro.

In alcuni casi si cerca di dividere i due aspetti, si usa una mano per muoversi, così si lascia l'utente libero di guardarsi intorno, introducendo però i problemi detti prima. Inoltre, se si usa la mano per ruotare la camera, significa avere in mano gli occhi, il che è una cosa strana, perché normalmente l'utente muove la testa. In quel caso è quindi necessario uno sforzo cognitivo, l'utente potrebbe muovere la mano e spostarsi senza che lo volesse, serve una coordinazione non semplice.

Il modo più semplice è il **teletrasporto**, l'utente indica dove andare, e si sposta tramite teleportation. Questo riduce anche il **cyber sickness**, è comodo ma non è realistico.



Per quanto riguarda la categoria del **camminare**, questa è difficile da fare nel VR, perché necessita di uno spazio enorme.

Un metodo è quello di trasformare piccoli movimenti nel mondo reale, in movimenti più grandi nel virtual world.

Altrimenti si usa il **walking in place**, facendo step sul posto, traducendo questi step in movimenti nel mondo virtuale.

Altrimenti si possono usare piattaforme fisiche come i **treadmills**.

Il **redirect walking** è una tecnica dove si fa in modo che la scena del virtual world sia manipolata a runtime, per spingere l'utente a cambiare direzione. Quindi si lascia che l'utente cammini, ma quando sta raggiungendo un'ostacolo, si fanno delle modifiche nel virtual world per allontanarlo dall'ostacolo del mondo reale. Questo va fatto a livello software, l'utente dovrebbe poter non accorgersi di questo trucco. Si può anche usare dei suoni per supportare questo meccanismo, facendo per esempio girare l'utente.

Le **leaning interfaces** sono degli oggetti fisici che funzionano come andare su sci, snowboard. L'utente è su una pedana e in base a come si inclina, se si sporge in avanti si muove, se torna dritto allora si ferma. Questo è utile perché lascia le mani libere però l'utente deve abituarsi. Un'esempio è la balance board della wii, ma ci sono oggetti più moderni. Le interfacce **isometriche** vanno a misurare per esempio la tensione muscolare, convertendola in movimento (come la balance board), mentre le interfacce **isotoniche** non hanno alcuna counterforce quando le si usa, non c'è nessuna resistenza. La combinazione di queste due è chiamata interfaccia **elastica**.

Ci sono quindi anche interfacce basate sull'inclinazione dell'utente, per esempio uno sgabello che funziona tramite i cambi di bilanciamento della

persona, inclinandosi.

C'è anche lo **speed teleporting**, dove al posto di scomparire e ricomparire da qualche parte, l'avatar si muove molto velocemente verso il posto selezionato. Questo però può reintrodurre la cyber sickness.

Nella **guided navigation** si usa un mezzo di trasporto virtuale (tipo una macchina, un trenino, un ascensore). Nelle tecniche di **route plan** o **goal-based movement** l'utente specifica un percorso verso il target, che è automaticamente seguito (questo metodo a 2 step non molto comune).