

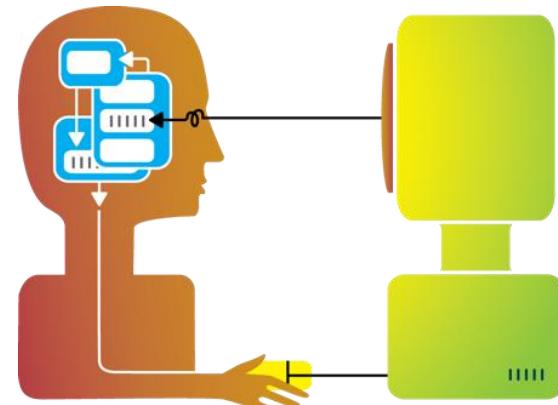


Einführung in die Spieleprogrammierung

User Interfaces



- User Interface
 - = Benutzerschnittstelle (Schnittstelle zwischen Mensch und Computer, Human-Computer Interaction = HCI)
 - Schnittstelle zum Nutzer über Ausgabe
 - Schnittstelle zum Computer über Eingabe





Hardware

- *Eingabegeräte*: Maus, Tastatur, Gamepad, Mikrofon, Kamera ...
- *Ausgabegeräte*: Monitor, Lautsprecher, ...
- *Ein-/Ausgabegeräte*: Touchscreen, Force-Feedback-Joystick, ...

- Ausgabe durch Rendern der Bildschirmanzeige (Text, virtuelle Umgebung, GUI), Generierung von Audio (Sprache, Musik, Geräusche), Erkennung von Kollisionen für ForceFeedback, ...
- Eingabe durch Verwendung der Daten aus den Eingabegeräten zur Cursor-, Kamera- oder Avatarsteuerung, Mapping von Tasten-/Button-Events auf interne Events, Spracherkennung, Gestenerkennung, ...



- Primäres Eingabe-Interface am PC ist üblicherweise Maus/Tastatur → Cursor- oder Kamera- bzw. Avatarsteuerung
 - Bei Konsolen traditionell Gamepads als Primäreingabe, grundsätzlich aber auch Genre-abhängig (z. B. Joysticks für Flugsimulatoren)
- Primäres Ausgabe-Interface am PC ist üblicherweise die Bildschirmanzeige
 - Unterscheidung zwischen *Shell-Interface* (Spieleinstellungen) und *In-Game-Interface* (Interface im eigentlichen Spiel)

Ubiquitous Computing

Embodiment

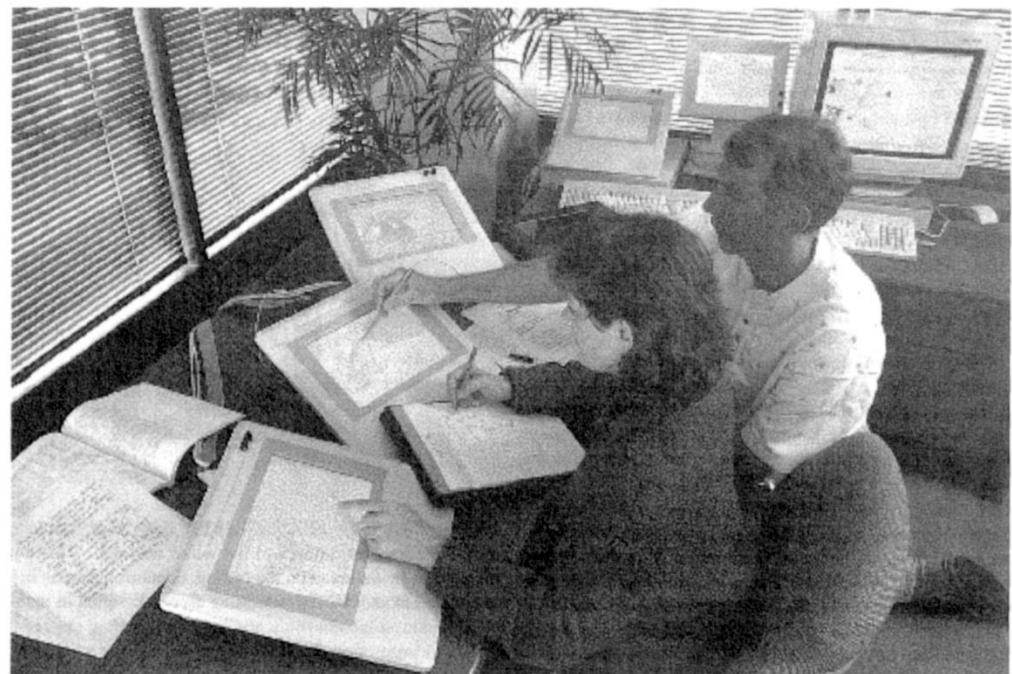


Figure 4: **COMPUTER SCRATCHPADS** augment the conventional screen in this office at the Xerox Palo Alto Research Center. Prototype pads are wired to conventional computers; thus far only a handful of wireless models have been built.



- Interface für das eigentliche Spiel (In-Game-Interface)
- Meist kein WIMP, sondern eigene Metaphern
- I. d. R. bezieht sich der Begriff nicht auf die Anzeige der Spielwelt selbst, sondern auf alle zusätzlichen Elemente der Anzeige (GUI)
- Unterscheidung zwischen:
 - Reine Ausgabe, (HUD=Head(s)-Up-Display)
 - Ein- und Ausgabe (Control-Interface)



HUD (*aber: Fadenkreuz dient auch der Eingabe*)



Control-Interface



Allgemeine Operationen:

- Navigation
- Selektion
- Manipulation
- Dialogauswahl
- Aber auch sehr spielspezifische:
 - Zielen
 - Einheiten gruppieren/formieren
 - Angriff befehlen
 - Heiltrank anwenden
 - ...

Evolution von In-Game-Interfaces

Zork (1977) <http://www.ifiction.org/games/index.php?cat=2>

```
>look
Kitchen
You are in the kitchen of the white house. A table seems to have been
used recently for the preparation of food. A passage leads to the west
and a dark staircase can be seen leading upward. A dark chimney leads
down and to the east is a small window which is open.

>look at passage
There's nothing special about the way.

>look in chimney
You can't look inside a chimney.

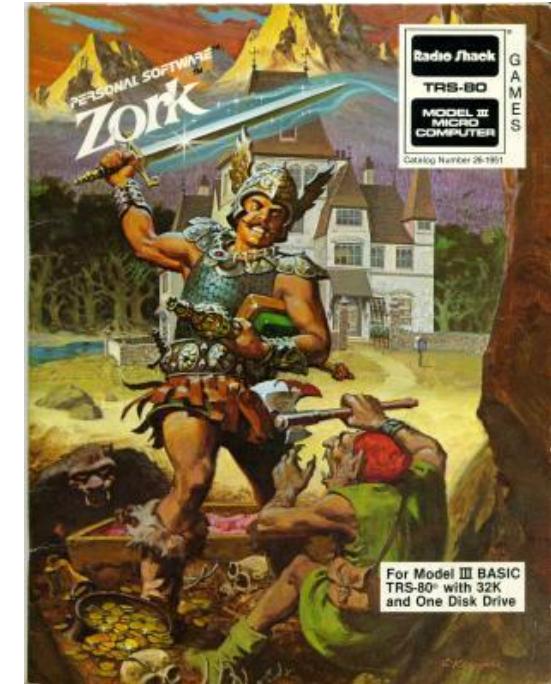
>go up stairs
You have moved into a dark place.
It is pitch black. You are likely to be eaten by a grue.

>use lantern
I don't know the word "use".

>light lantern
The brass lantern is now on.

Attic
This is the attic. The only exit is a stairway leading down.
A large coil of rope is lying in the corner.
On a table is a nasty-looking knife.

>
```



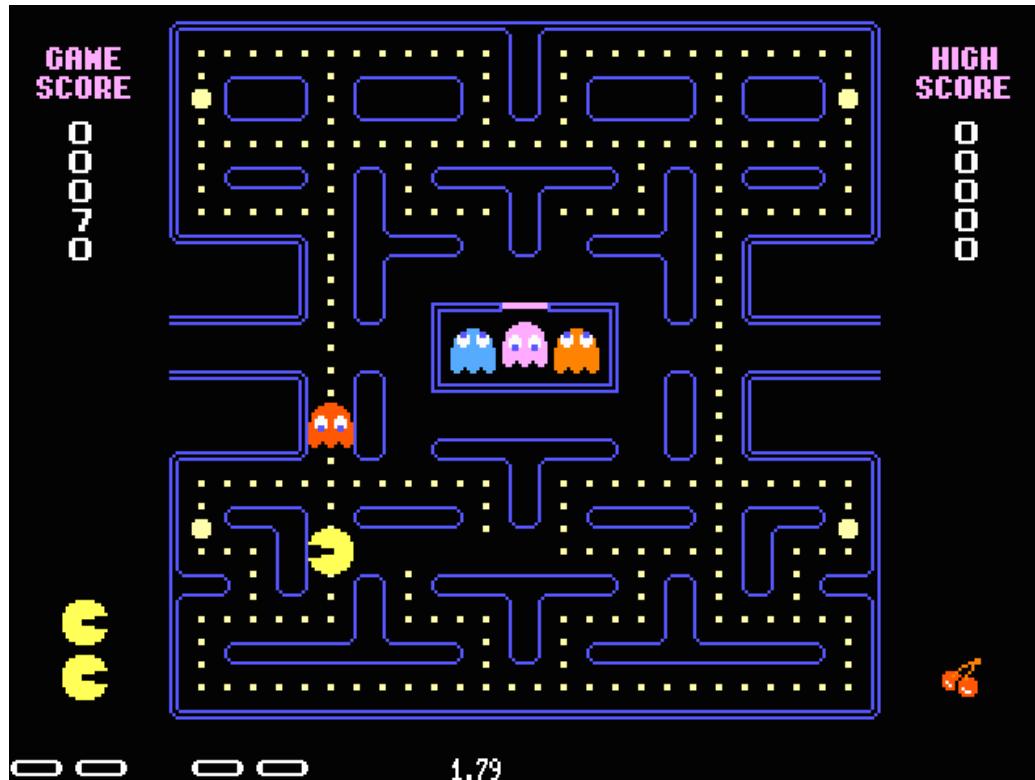
Reine Text-basierte Ein- und Ausgabe

Evolution von In-Game-Interfaces



Mystery House (1980): Text-basierte Ein- und Ausgabe + Analoggrafik

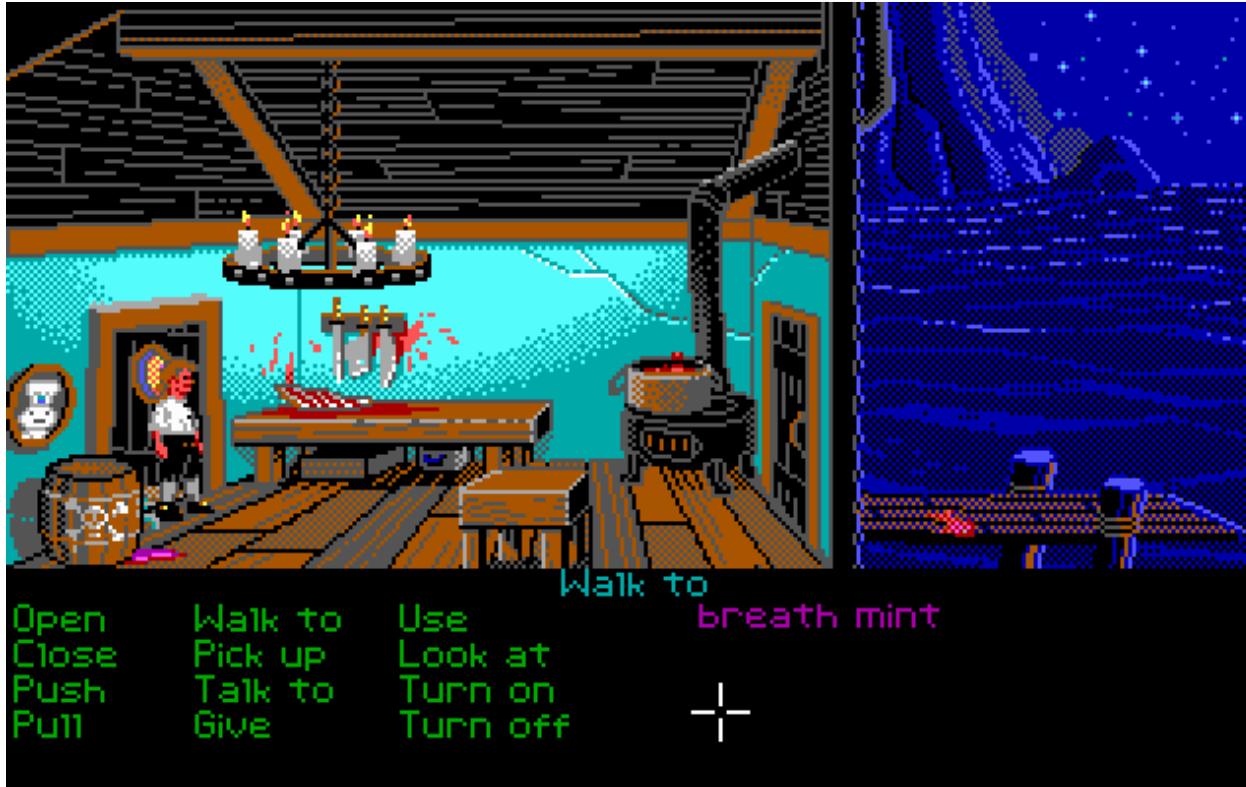
Evolution von In-Game-Interfaces



Pac-Man (1980): Text+Icon HUD mit Sprites, Eingabe über Joystick

Auch die ersten (3D) First-Person Shooter (*rechts*: Catacomb 3-D, 1991) hatten ein Text+Icon HUD

Evolution von In-Game-Interfaces



Diverse Adventures (hauptsächlich Sierra und LucasArts) ab Mitte der 1980er fügen zunächst Grafik hinzu und ersetzen die Texteingabe durch die Maus
Hier: The Secret of Monkey Island (1990): Maussteuerung (Point-and-Click) mit Text-GUI basierend auf 12 Aktionsverben + Inventar

Evolution von In-Game-Interfaces



King's Quest V (1990): Aktionsverben werden mit GUI Icons ersetzt

Evolution von In-Game-Interfaces



„Moderne“ Adventures (*links*: Geheimakte Tunguska, 2006; *rechts*: Geheimakte 3, 2012): Point-and-Click Steuerung mit Aktionsoptionen direkt am Mauszeiger. Icons mit direkter Visualisierung für das Inventar.

Evolution von In-Game-Interfaces



Strategiespiele (*links*: Dune II, 1992, *rechts*: Age of Empires II, 1999) entwickelten ein komplexes Control-Interface mit Maussteuerung und verschiedenen Elementen: Icons, Text, Statusbalken, Übersichtskarte, ... Teilweise in die Spielwelt integriert

Evolution von In-Game-Interfaces



World of Warcraft (2004): Durch Benutzer anpassbares Interface

Evolution von In-Game-Interfaces



Radiale Auswahldialoge
Mass Effect 3 (2012)
Crysis (2007)

Evolution von In-Game-Interfaces



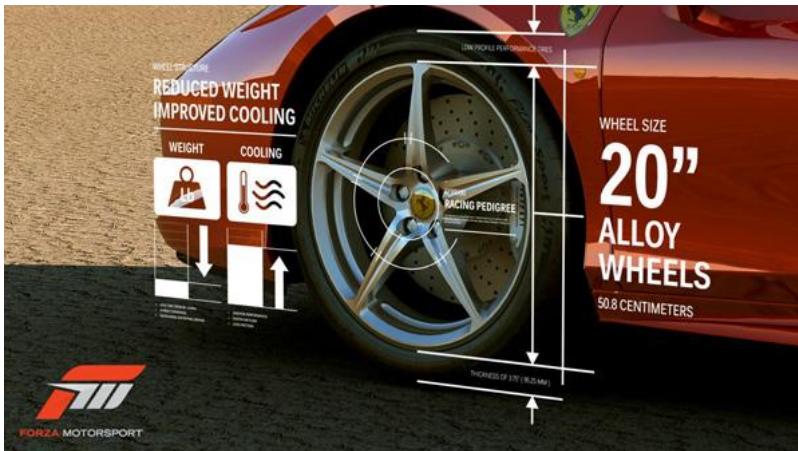
Gesundheit Stasis
(= „Energie“)



Hinweise für Spieler
als Projektion in die
Spielwelt

Vollständige Integration der Anzeigen in die Spielwelt
Dead Space (2008)
Splinter Cell Conviction (2010)

Evolution von In-Game-Interfaces



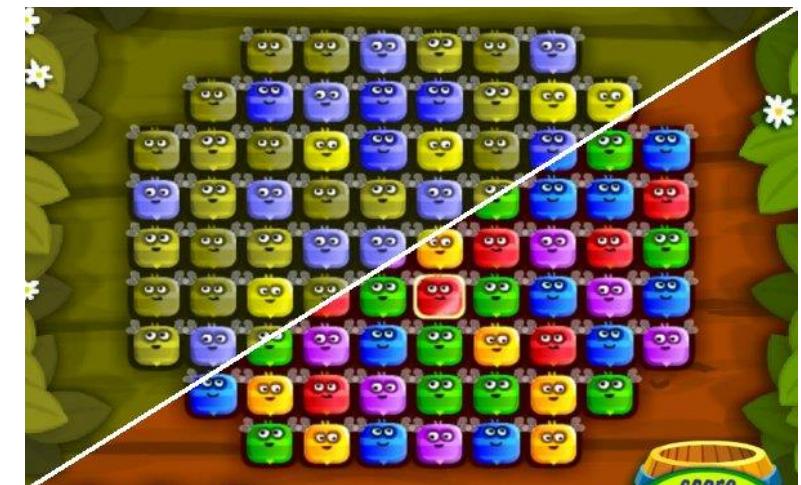
Integration der Anzeigen in die Spielwelt als „Augmented Reality“
Forza Motorsport 4 (2011)
Batman: Arkham City (2011)



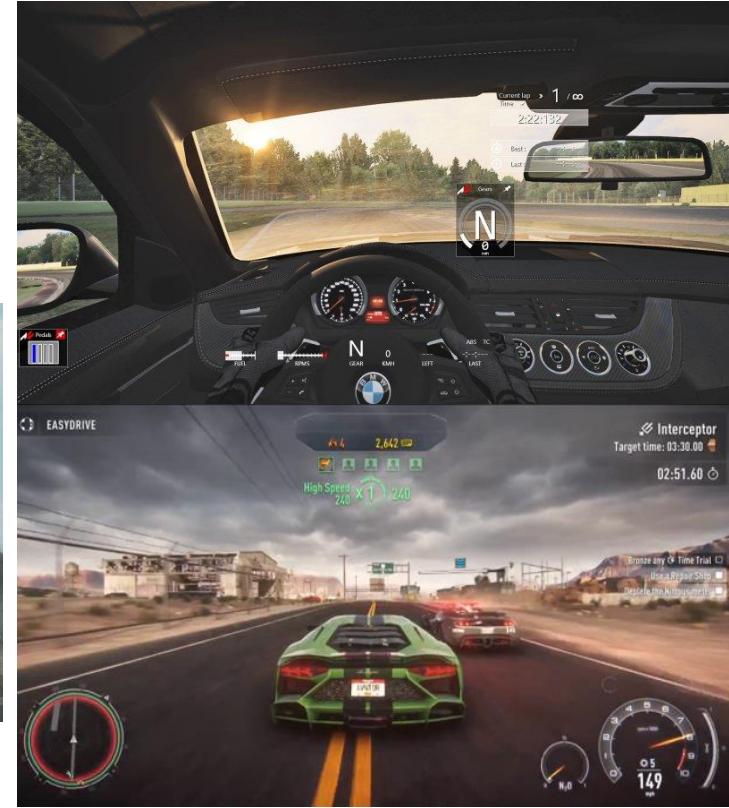
Wie erschafft man ein intuitiv bedienbares In-Game-Interface?

- Erforschbarkeit und Erlernbarkeit
- Effizienz und Effektivität
- Konsistenz und Standards
- Einfachheit
- Immer Anzeigen was gerade passiert und wichtig ist
- Verwendung von Metaphern/Affordanzen
→ Icons (aber mit Vorsicht)
- Feedback/Feedforward Mechanismen
- Möglichkeit etwas zu pausieren/abzubrechen/rückgängig zumachen → Speichermöglichkeiten

- Benutzerführung bzw. Begrenzung der Auswahlmöglichkeiten
- Immersion aufrecht erhalten
- Vorhandenes Wissen nutzen
 - Wissen über Realität
 - Computer-Wissen
 - Basis-Spiel-Wissen
 - Genrespezifische Vorgaben
- Interface durch Benutzer anpassbar machen
- Unterschiede/Einschränkungen der Nutzer beachten
 - Links-/Rechtshänder, Rot-Grün-Blindheit, Stereoskopie-Blindheit, kulturelle Unterschiede, ...
- Ästhetik bzw. Look and Feel
- „Easy to learn, hard to master“
- ...



In-Game-Interface Design



Simulation oder Action?

Falcon 4.0 (1998) vs. Tom Clancy's Hawx 2 (2010)

Assetto Corsa (2014) vs. Need for Speed: Rivals (2013)

Nachgebildetes Original-Interface bietet zwar mehr Authentizität,
überfordert aber insbesondere Gelegenheitsspieler

Verwendung von (genrespezifischen) Standards



Monkey Island



Zak McKracken



Baldur's Gate



Diabolo

Verwendung bekannter Interface-Elemente

Hier: SimCity 2000 (1994)

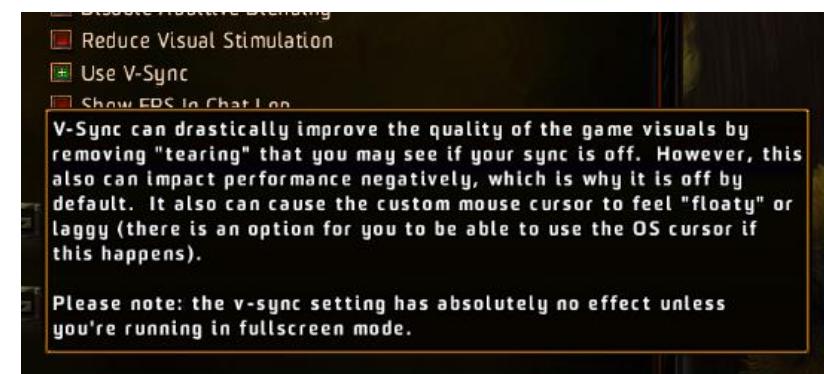


- Wo es passt bekannte Mausaktionen, Menüsysteme oder Dialogfelder, Drag-and-drop, Scrollbars etc. verwenden

- Eher Grafiken als Text verwenden:
 - Grafiken sind sprachunabhängig
 - ...einfacher zu verarbeiten
 - ...brauchen oft weniger Platz
 - ...sehen besser aus
- Zu beachten:
 - aussagekräftige Icons verwenden
 - einfache Bilder sind leichter zu verarbeiten
 - Farben gezielt einsetzen
 - einfach zu erkennende / sich unterscheidende Formen verwenden
 - Unterschiedliche Bedeutung in verschiedenen Kulturen beachten
 - Icons benötigen meist eine gewisses Einlernen → Tutorial/Einführung, Während dem Spiel: Tooltips
 - Schlecht gewählte / zu viele Icons können ein Interface auch unnötig kompliziert machen



Interface „erforschbar“ halten
→ Tooltips erklären Elemente näher





Verwendung von Standards bei der Eingabe, z. B. bei Tastenbelegung/Mausverwendung:

- WASD-Steuerung für Fortbewegung
- Leertaste: Springen
- R: Nachladen
- E oder F: Interagieren
- STRG oder C: Ducken
- ESC: Abbrechen, Pausieren, Hauptmenü
- Zahlentasten: Waffe wechseln oder Einheitengruppen auswählen
- F5/F6 oder zwei andere neben einander liegende F-Tasten: Quicksave/-load
- ...

- Kamera-/Avatar-Rotation ändern durch Mausbewegung
- Maussteuerung um Einheiten/Gebäude zu selektieren u. Ziele zu markieren in Strategiespielen
- Mausrad um Waffe zu wechseln
- Maus über Element: Highlighting, Tooltip
- Linksklick: Auswählen; Rechtsklick: Kontextmenü
- Maustaste halten → Drag and Drop
- ...

Geringe Distanz zwischen zusammengehöriger Ein- und Ausgabe → Verwendung von Kontextmenüs



- Die Entscheidungszeit T bei mehreren Alternativen steigt proportional zum Informationsgehalt H

$$T = b \cdot H$$

Bei gleichverteilter Wahrscheinlichkeit:

$$H = \log_2(n + 1)$$

$$T = b \cdot \log_2(n + 1)$$

Allgemeiner Fall:

$$H = \sum p(i) \log_2(1 / p(i) + 1)$$

T: Entscheidungszeit

H: informationstheoretische Entropie einer Entscheidung

b: Nutzerabhängige Konstante, oft als 150 msec angenommen

n: Anzahl gleichwertiger Alternativen

p(i): Wahrscheinlichkeit für Alternative i

→ *Übliche Regel:* Nicht mehr als 7 Optionen ≈ 0.5 sec Entscheidungszeit

- Die Aktionszeit (für eine Selektion mit der Maus in der GUI) steigt mit zunehmender Entfernung und abnehmender Größe des Ziels.

$$t = a + b \cdot \log_2 \left(\frac{D}{W} + 1 \right)$$

t: Zeit um Ziel zu erreichen

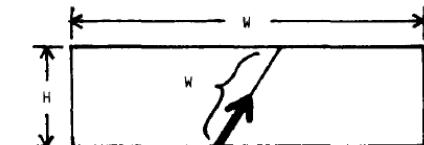
D: Entfernung zum Ziel

W: Größe des Ziels

a, b: Parameter abhängig vom Eingabegerät



Erweiterung von McKenzie (1992) für 2D:



→ Zusammengehörige Controls nahe beieinander, nicht zu klein, evtl. auf Form achten

Immersion bewahren = „Herausreißen“ aus dem Spiel
möglichst vermeiden

→Overlays vermeiden,
stattdessen Anzeigen
direkt in die Spielwelt
integrieren

Hier: Halo 3 (2007)



In-Game-Interface Design

Auch Shell-Interface, Inventar oder Menüs zum verteilen von Erfahrungspunkten können direkt in die Spielwelt integriert werden, z.B. GTA 4 (2008), Fallout 3 (2008), Alone in the Dark (2008)



In-Game-Interface (not) Design



Hardware

Ausgabegeräte

- Primäre Ausgabe meist über Monitor mit verschiedenen Techniken:
 - Röhre, Plasma (PDB), (TFT-)LCD/LED, (AM)OLED oder auch Beamer mit Leinwand
 - *Wichtige Eigenschaften:* Größe (Diagonale), Bildwiederholrate, Auflösung (Anzahl Pixel und Seitenverhältnis, aber auch Pixeldichte), Farbanzahl, Reaktionszeit, Kontrastverhältnis, 3D-fähigkeit, Synchronisation (VSync, Gsync, FreeSync), ...
- Sekundäre Ausgabe über Lautsprecher/Kopfhörer: Stereo vs. Raumklang, Spiele liefern i.d.R. 3D-Sound
- Sonstige Ausgaben: Force-Feedback durch Controller, LEDs am Controller, ...



HMDs bzw. VR Brillen



- Headmounted Display → Brille mit Display(s) direkt vor dem Auge
- Kleine Displays, aber durch die Nähe groß wie eine Leinwand
- Hohe Immersion (Eintauchen in die virtuelle Welt)
- Meist mit Kopfhörern
- Teilweise mit Head Tracking → Kopforientierung/-position → Steuerung d. virtuellen Kamera → Noch höhere Immersion
- Teilweise mit „Außenkamera“ bzw. See-through oder Tiefenkamera → Augmented Reality
 - Vgl. z. B. Microsoft HoloLens



Probleme bei HMDs

- Zu kleines Sichtfeld
 - Bei früheren VR Brillen üblich ca. 30° diagonal bzw. 65-Zoll Display gesehen von 3 m Entfernung
 - Sichtfeld des Menschen ca. 180° horizontal und ca. 130° vertikal, fixierbar maximal 60° horizontal, 75° vertikal
- Headtracking mit zu großer Verzögerung → „Nachruckeln“ der virtuellen Kamera
- *Anfang 2016:* Oculus Rift verspricht deutlich größeres Sichtfeld (110° diagonal ermöglicht durch spezielle Linsen + Renderingtechniken zum Ausgleich von Verzerrungen) und weniger Latenz (1000 Hz) beim Tracking
- *Ähnliche Konkurrenz:* Sony Morpheus mit Move-Controllern und Steam VR mit neuen Controllern



- Stereoskopie über HMDs
- Stereoskopie über Monitor bzw. Beamer/Leinwand
 - Aktive/Passive Brillen
 - Autostereoskopie ohne Brille
- Nutzung für Interfaces:
 - Realistischere Wahrnehmung der virtuellen Welt, höhere Immersion
 - 3DUIs
 - *Vorsicht:* ~10% der Menschen können Stereoskopie nicht/nur eingeschränkt wahrnehmen, mindestens 30% haben Probleme damit
→ Möglichst optional gestalten







Eingabegeräte (Game Controller)

- Anforderungen an Game Controller:
 - Stabilität/Robustheit (oft hohe Beanspruchung)
 - Kosten (vor allem bei spielspezifischen Controllern)
 - Wiederverwendbarkeit/Universaliät (Verwendung in verschiedenen Spielen/Genres)
 - Ergonomie
 - Präzision/Reaktionszeit
 - Fehlertoleranz
 - Individuelle Anpassbarkeit (Linkshändermaus, Zusatzgewichte für Maus, Anpassung von Geschwindigkeit, Key-/Buttonmapping)

Tastatur

- Standardeingabegerät für Computer
- Schablonen
- Spezielle Keypads
- Texteingabe
- WASD-Steuerung
- Benutzung von Shortcuts und Makros



Maus

- Standardeingabegerät für Computer
- Hohe Präzision
- Bedienung von UI Elementen mit Cursor
- Rotation von Kamera/Avatar
- *Auch:* Mausgesten



Black & White (2001)



Ⓐ Normale Wunder



◎ Kreaturwunder



Joystick

- Standardeingabegerät der „Ur-Konsolen“
- Digital (An/Aus) i.d.R. vier Richtungen
- Analog (Intensität abhängig von Auslenkung)
- Verwendung in Flugzeugsimulatoren, aber auch allgemein zur Navigation/Rotation und Cursorsteuerung
- Oft mit Force-Feedback



Datenhandschuh und Armband

- Datenhandschuh
 - Analoge Messung der Fingerkrümmung
 - Manchmal auch Messung von Kontaktpunkten
 - Auch kombiniert mit 3D Tracking der Hand im Raum, Bewegungssensorik, oder haptisches Feedback über Vibration
- Myo Armband (2015)
 - Analysiert Muskelaktivität, zur Erkennung von Fingergesten/-posen
 - Zusätzlich auch Bewegungssensorik und haptisches Feedback über Vibration



Gamepad

- Standardeingabegerät für Konsolen
- Oft mit Force-Feedback
- Aktuelle Gamepads kombiniere meist Buttons mit digitalen und analogen Joysticks
- Inzwischen aber auch Kameras, (Multitouch-) Displays, Lautsprecher, Bewegungssensorik, ... → Eigentlich schon eigenständige Handheld-Konsolen
- „Allround-Eingabegerät“



Gamepad

Brothers: A Tale of Two Sons



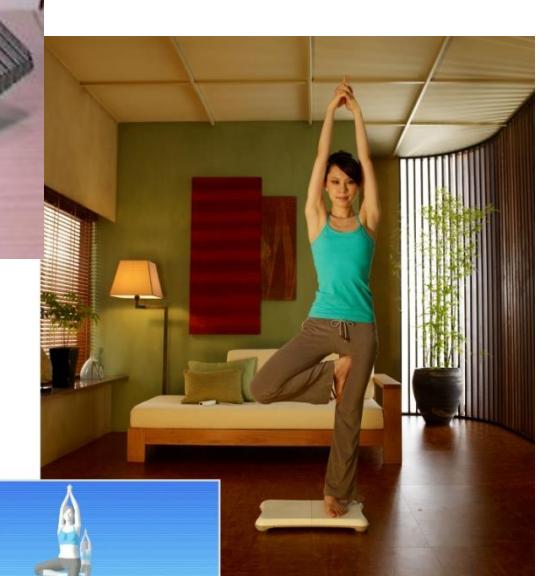
Lenkrad

- Auch mit Pedalen und Schaltknüppel
- Oft mit Force-Feedback
- Zur realitätsnahen Fahrzeugsteuerung



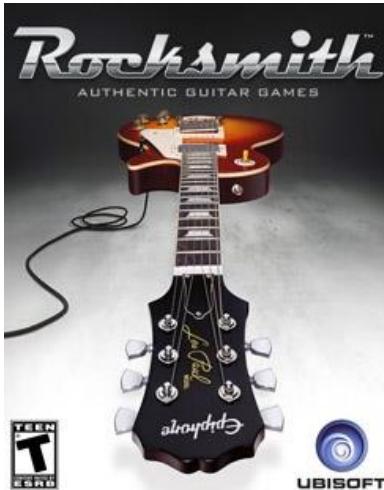
Tanzmatte und Board

- Tanzmatten und Joyboards verwenden Ein/Aus-Taster
- Wii Balance Board (2007) verwendet vier analoge Drucksensoren
- Verwendung für Tanz- und Fitness-Spiele, aber auch zur Navigation/Richtungsangabe

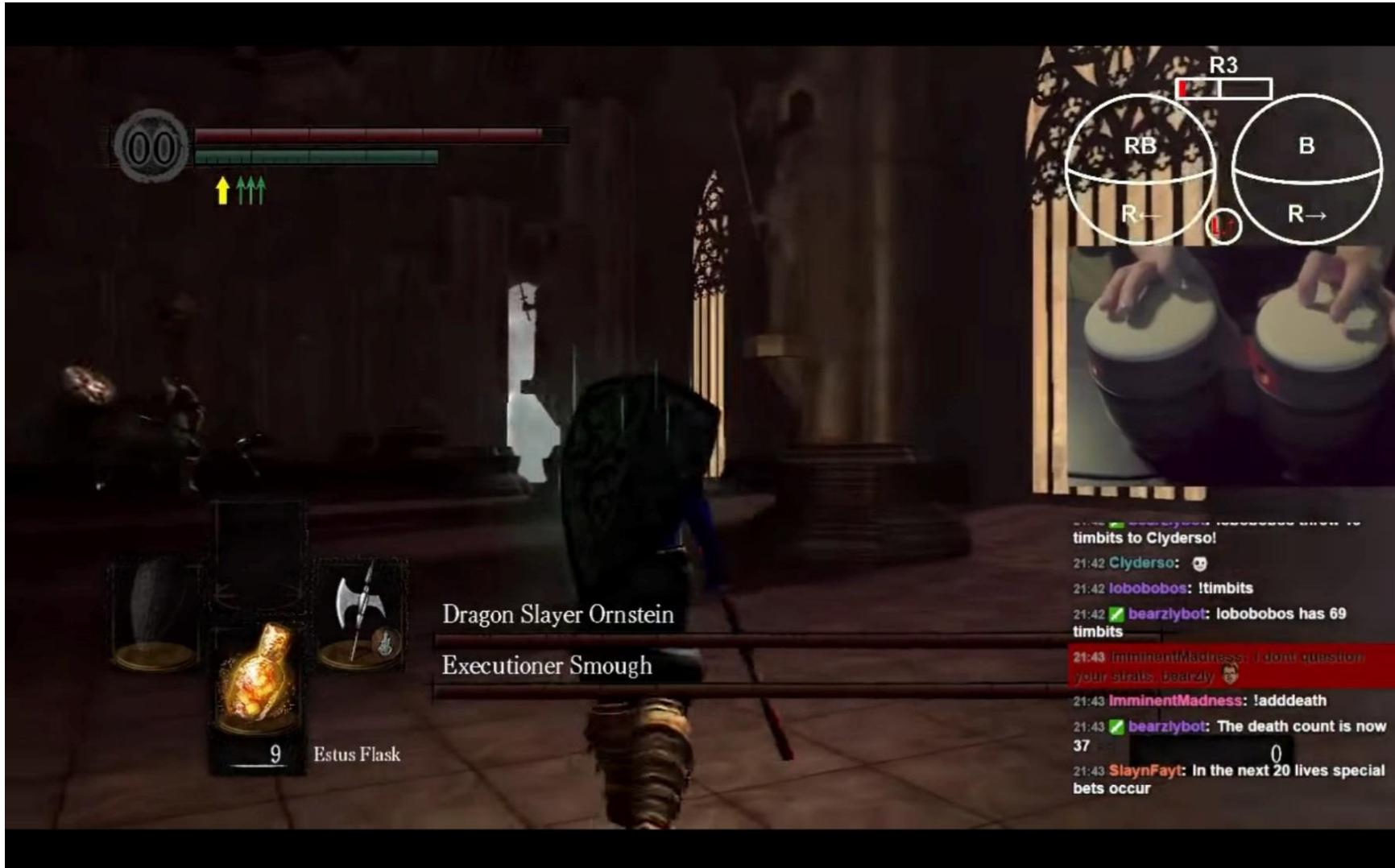


Instrument

- Donkey Konga (2003) und Taiko Drum Master (2004)
- Guitar Hero (2005) zuerst nur mit Gitarre
- Rock Band (2007) mit Mikrofon und Schlagzeug
- Inzwischen auch Rocksmith (2011) mit echter Gitarre

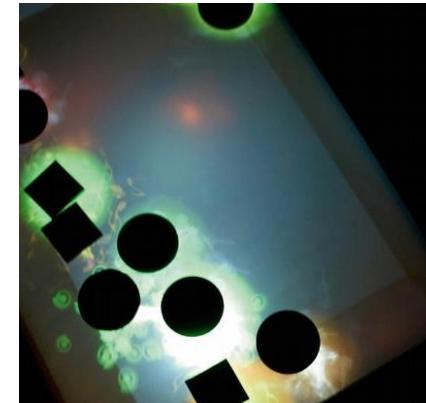
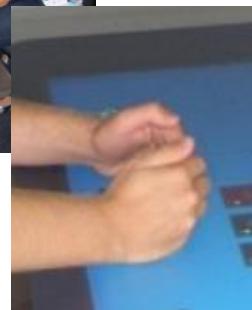
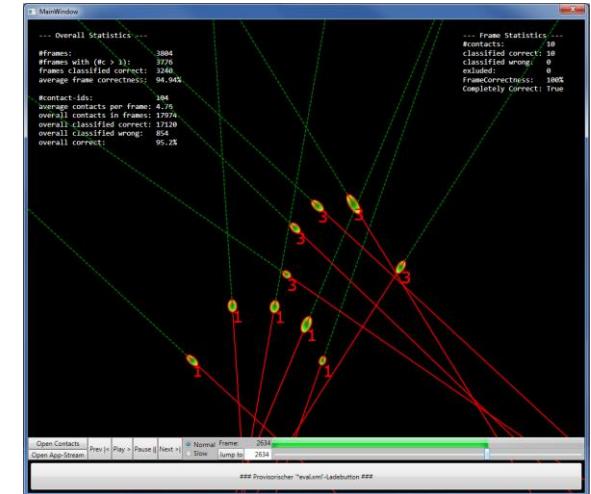


Instrument



Touchscreen

- Erkennung von Fingerpunkten, Handposen, aber auch Objekten (Tangibles)
- Nischendasein von entsprechenden (PC-)Spielen wegen Hardware-Beschränkung



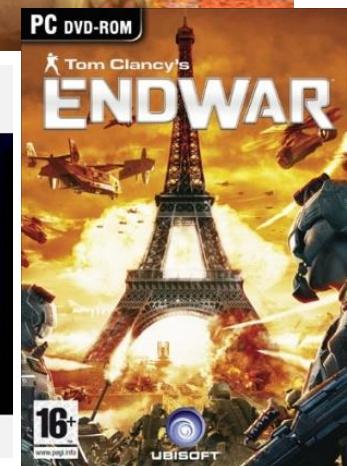
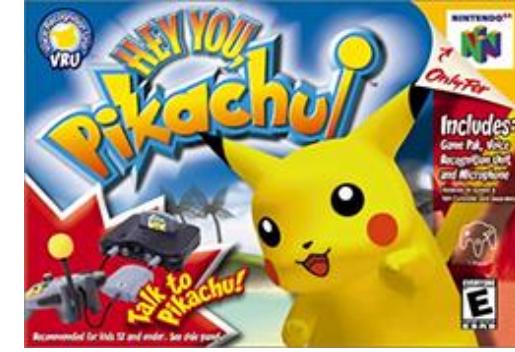
<http://hcm-lab.de/projects/SheepMeUp/>

Touchscreen

- Mobile Geräte mit Touchscreen
 - eigentlich alle aktuellen Handheld-Konsolen (Nintendo 2/3DS, PS Vita, Pandora, ...), sowie Smartphones und Tablets
 - *Auch:* Verknüpfung mit externem Display (Wii U)



- Kommandobasiert:
 - Hey You, Pikachu! (1998)
 - Nintendogs (2005)
 - Tom Clancy's EndWar (2008)
- Chat:
 - Teamspeak etc.
- Stimmbasiert:
 - Karaoke
 - Emotionserkennung



- Farbkamera

- EyeToy (2003) bzw. Playstation Eye (2007)
 - Farbkamera/Mikrofon
 - Hand-/Gesichtserkennung, Markererkennung für Interaktion, (Indirect) Augmented Reality, ...
- Eye of Judgment (2007)
("turn-based card battle video game")
- EyePet (2009)
 - iPad Release 2010!



Kamera

- Tiefenkameras

- Microsoft Kinect



- Ganzkörpertracking und Gesten-/Posenerkennung

- Leap Motion Sensor

- Ermöglicht Fingertracking durch kleine Kamera unten, vor dem Bildschirm



- Eigener „Airspace Store“ auch für Spiele



Eye oder Head Tracker

- Head Tracker erkennt Kopfposition und/oder -orientierung
 - Erkennung über Bewegungssensoren am Kopf oder Tracking durch Kamera(s)
 - natürlichen Kamerasteuerung für HMDs (Oculus Rift, etc.)
 - Position für Pseudo-3D-Effekt bei Monitoren
- Eye Tracker erkennt wohin der Nutzer sieht und was er dabei fixiert
 - Ersatz für Mauszeiger, aber wie kann man „klicken“? → oft über Verweilen (engl. dwell)



Eye Tracker

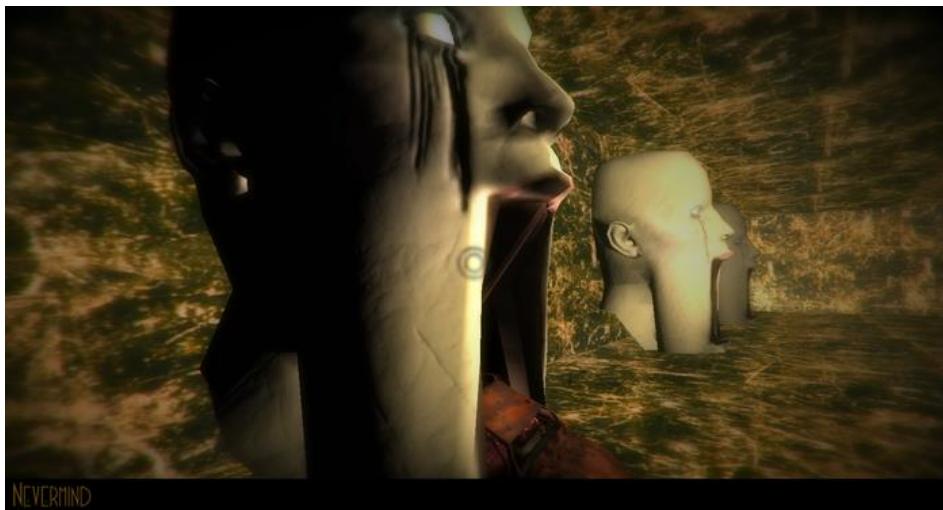


Get absorbed by Soma's frightful ambience by triggering different effects throughout its game world with your eyes

Biosensor

- Meist verwendete physiologische Signale:
 - SC (Skin Conductance = Hautleitwert)
 - BVP (Blood Volume Pressure = Blutdruck/Puls)
 - EMG (ElectroMyoGram = elektr. Muskelaktivitätskurve)
 - ECG (ElectroCardioGram = Herzspannungskurve)
- SC, BVP, ECG eignen sich, um z.B. Nervosität oder Erregung zu messen
- Mit dem EMG kann man z.B. Gesten erkennen (s. Myo)





NEVERMIND

Nevermind – Horror
BVP/Heart-Rate



The Journey to Wild Divine –
Serie von Adventure-
ähnlichen Spielen zur
Stressbewältigung; benutzt
BVP + SC

- Brain-Computer-Interface (BCI)
- z.B. Emotiv EPOC
 - misst Gehirnströme über Elektroenzephalografie (EEG)
 - 16 Sensoren
 - 2 Gyroskope
- *Problem:* Gehirnströme werden oft durch Muskelaktivitäten im Gesicht überlagert



Bewegungssensorik

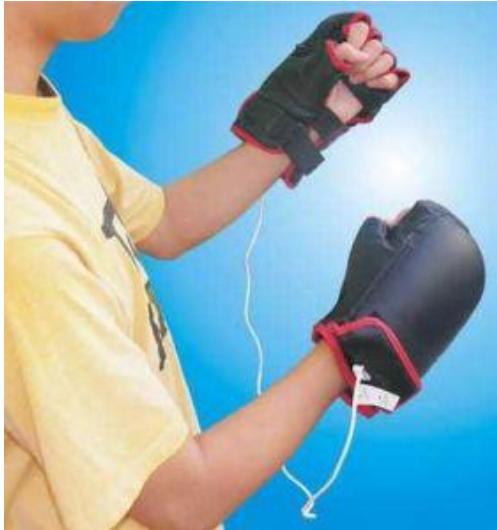
- Diverse Sensoren (meist in Kombination):
 - Beschleunigungssensor (Geschwindigkeit/Position über Ableitung)
 - Gyroskop (misst Winkelgeschwindigkeit, Orientierung über Ableitung)
 - Magnetometer (Orientierung anhand des Erdmagnetfeldes wie beim Kompass)
 - Aber auch Kameratracking (Wiimote: Infrarot, Sony Move: RGB)
- Erkennung von Position, Zeige- und Bewegungsrichtung, sowie Gesten zur Interaktion



Sonstiges



Sonstiges (2)

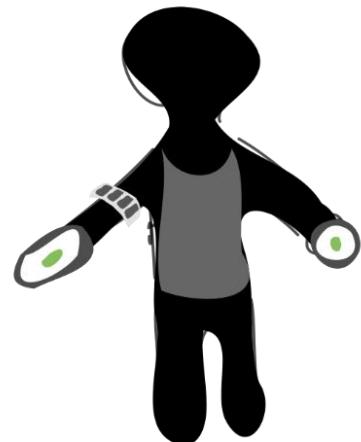


Mobil



Eingabe:

Touch
Mikrofon
Bewegung
Licht
Nähe
GPS
Kommunikations Logs..



Eingabe:

Mikrofon
Touch
Bewegung
BVP



Ausgabe:

Bildschirm
Lautsprecher
Vibration

- Mobile Spiele:
 - Gamification
 - Im Alltag
- AR
 - Pokemon Go
 - Ingress (Niantic)
 - Minecraft Earth

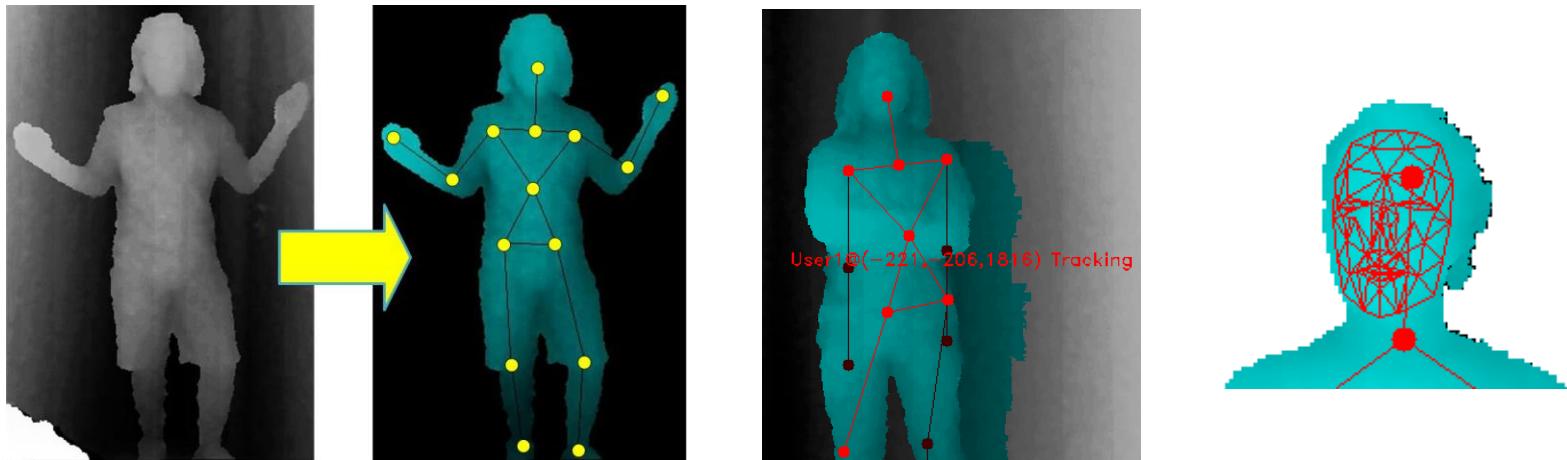


OpenNI™

NiTE™

KINECT™
for Windows®

- OpenNI/NiTE oder das Kinect for Windows SDK ermöglichen es Benutzer im Tiefenbild zu verfolgen (tracken) und ein vereinfachtes Skelett auf sein Silhouette im 3D-Raum zu legen; *bei Kinect SDK:* zusätzlich Facetracking möglich; OpenNI/NiTE werden seit dem PrimeSense-Kauf von Apple nicht mehr weiter entwickelt
- Relativ stabiles Tracking, hat aber auch einige bekannte Probleme mit nahen Wänden, Händen nahe dem Körper und Verdeckungen



- Im Gegensatz zu PDAs/Maus/Multitouchscreens/Webcam: 3D anstatt 2D Punkte
- Im Gegensatz zu PDAs/Maus/Wiimote: 15-20 Gelenkpunkte anstatt 1 Referenzpunkt
- *Aber auch:* Interaktion ständig aktiv, Daten vergleichsweise unpräzise und unverlässlich