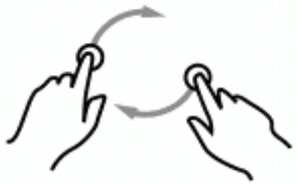
Game Experience Design

07 Interfaces

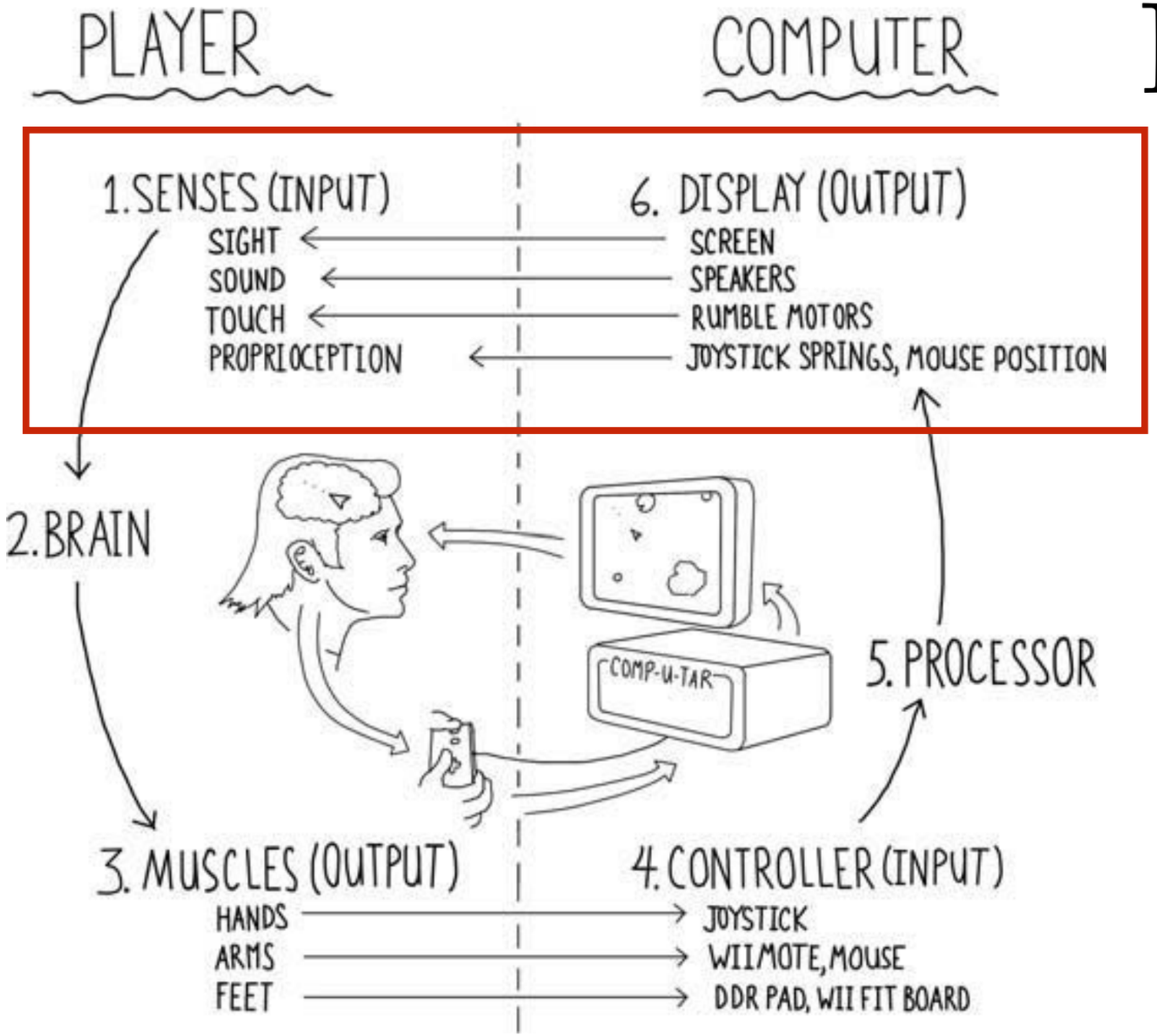
Prof. Dr. Jochen Koubek

13.06.2017

Interface-Design

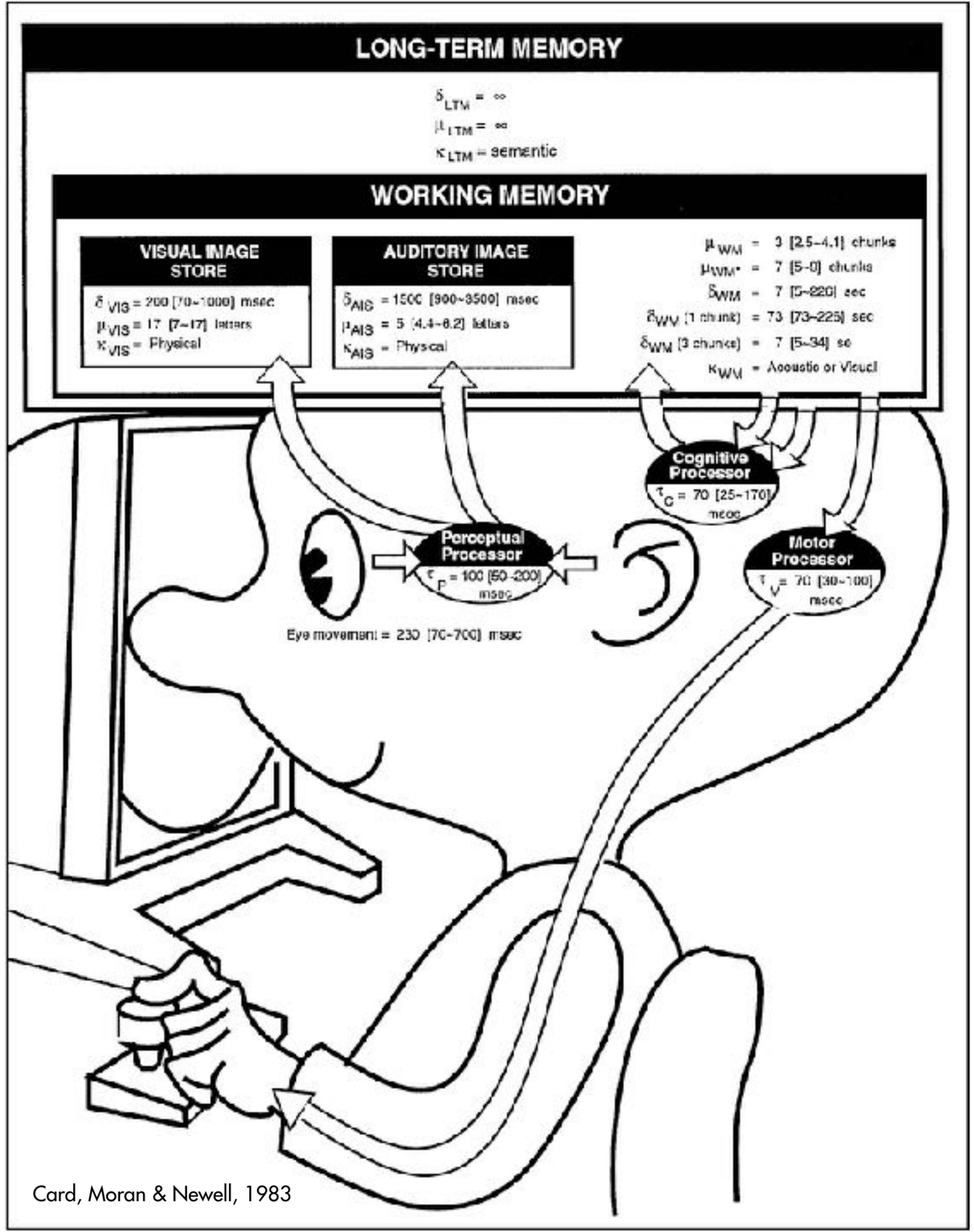


Feedback-Loop



s. 08 Multisensory Design

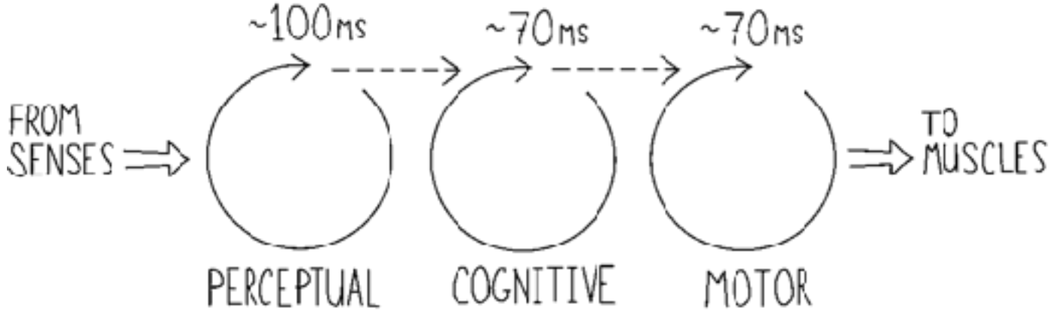
Swink: Game Feel, S. 34 f.



Card, Moran & Newell, 1983

Brain: Human Processor Modell

Average time required for the player to perceive the state of the game world and react to it is around 240 ms.



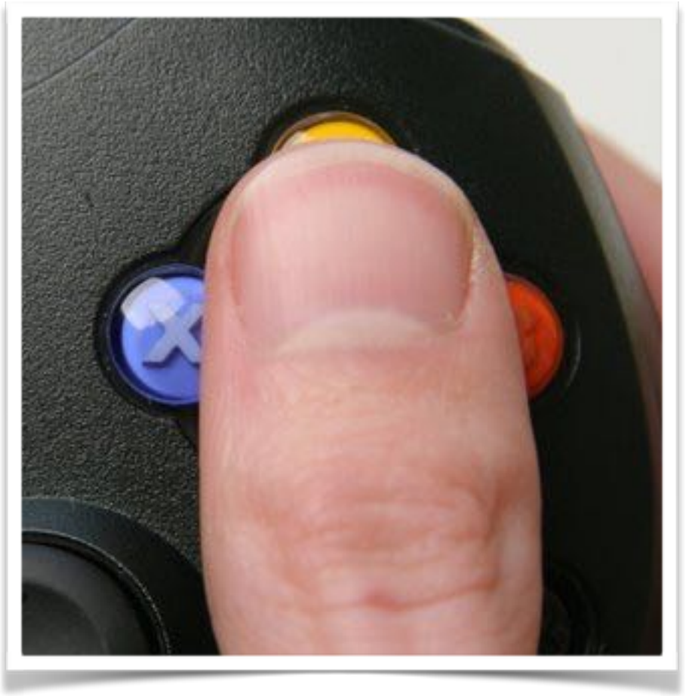
for the game controls to be comfortable, the system should have a response time no more than 100 ms.

https://www.humanbenchmark.com/

Output: Precise und Sloppy Players

http://cowboyprogramming.com/2007/01/02/pushhing-buttons/

The "precise" player rest the tip of their thumb over the primary button.

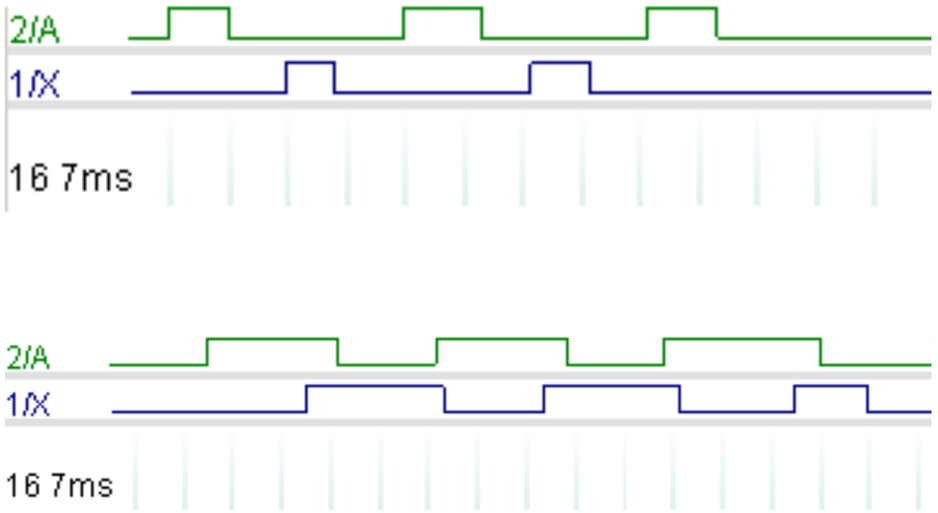


the "sloppy" player rests his thumb over all the buttons - allowing a button press by just tilting the thumb..

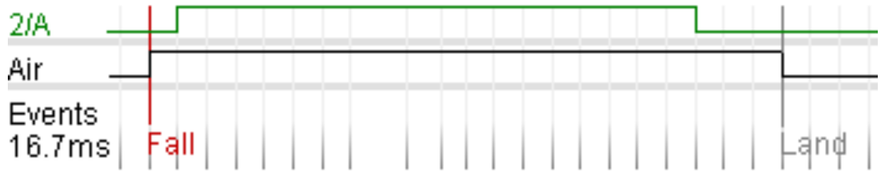
The Nintendo Gamecube controller is built around the expectation that the player will use some kind of "sloppy thumb" technique. There is a large central primary button, and with the three other buttons surrounding it, encouraging you to keep your thumb squarely over the primary button, and hit the other buttons with the edges and the tip of your thumb.

Verwechslungen

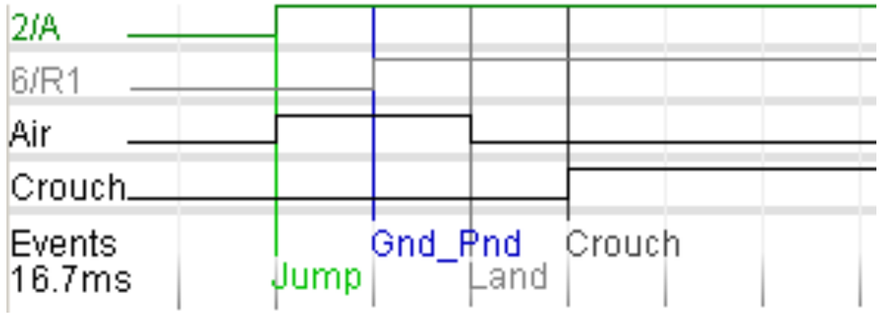
http://cowboyprogramming.com/2007/01/02/pushhing-buttons/



Timing graph for tapping A and X with Precise Thumb (top) and Sloppy Thumb (bottom)



Showing a jump attempt that failed by just 0.016 seconds.



The player tries to super jump by crouching then jumping, but accidentally jumps one frame before crouching causing an instant ground pound.

Controller-Hardware

Klassifikation von Eingabegeräten

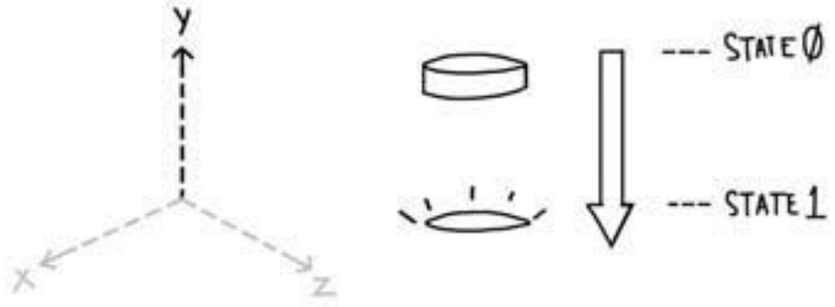
nach Jacob (1996): The Future of Input Devices

Space Invaders Controls

Hand – Diskret

Knöpfe

Tasten



Tastatur

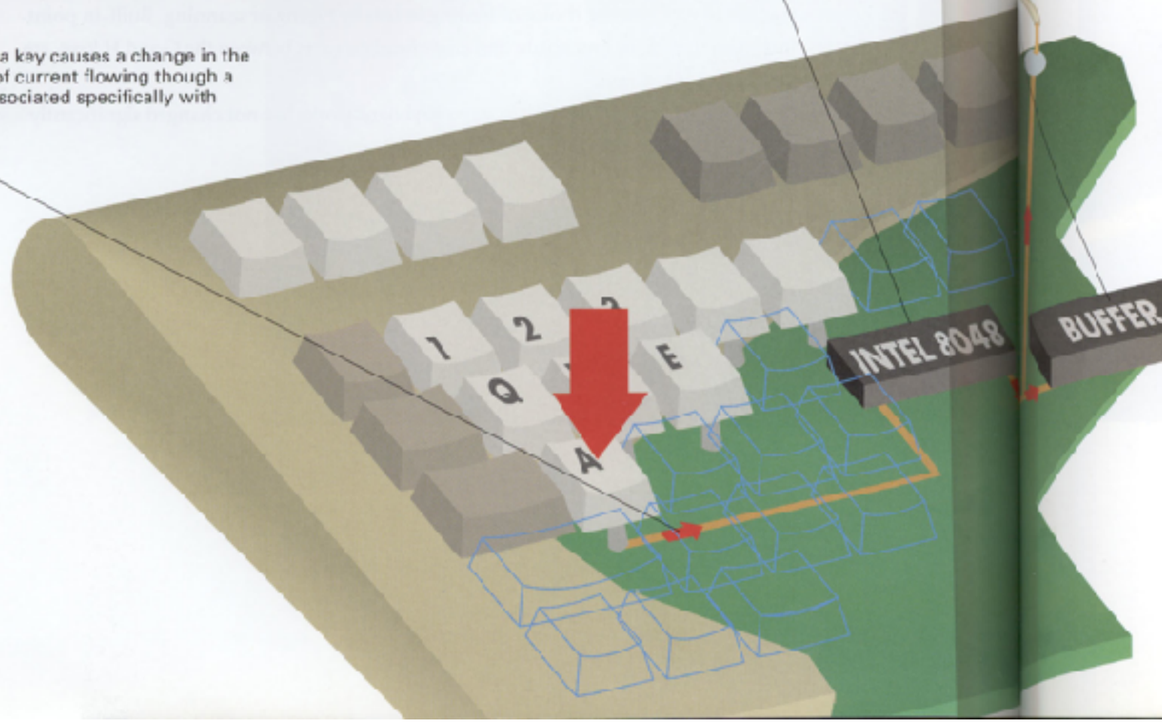

CHERRY
Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Druck
Rollen
Pause
Num
Rollen
Einfg
Pos1
Bild
Entf
Ende
Bild
Num
Pos1
Ende
Einfg
Entf
Bild
Enter
Strg
Alt
Alt Gr
Strg

The Keyboard and Scan Codes

3 Depending on which key's circuit carries a signal to the microprocessor, the processor generates a number, called a *scan code*. There are two scan codes for each key, one for when the key is depressed and the other for when it's released. The processor stores the number in the keyboard's own memory buffer, and it loads the number in a port connection where it can be read by the computer's BIOS (basic input/output system). Then the processor sends an interrupt signal over the keyboard cable to tell the processor that a scan code is waiting for it. An interrupt tells the processor to drop whatever else it is doing and to divert its attention to the service requested by the interrupt. (See Chapter 3.)

2 A microprocessor built into the keyboard, such as the Intel 8048, constantly scans circuits leading to the keys. It detects the increase and decrease in current from the key that has been pressed. By detecting both an increase and a decrease in current, the processor can tell when a key has been pressed and when it's been released. Each key has a unique set of codes, even if, to the users, the keys seem identical. The processor can, for example, distinguish between the left and right shift keys. To distinguish between a real signal and an aberrant current fluctuation, the scan is repeated hundreds of times each second. Only signals detected for two or more scans are acted upon by the processor.

1 Pressing a key causes a change in the amount of current flowing though a circuit associated specifically with that key.



SCAN CODE TABLE	
1E	A
30	B
2E	C

Scan code

1E

BIOS

BIOS

4 The BIOS reads the scan code from the keyboard port, and sends a signal to the keyboard that tells the keyboard it can delete the scan code from its buffer.

RAM

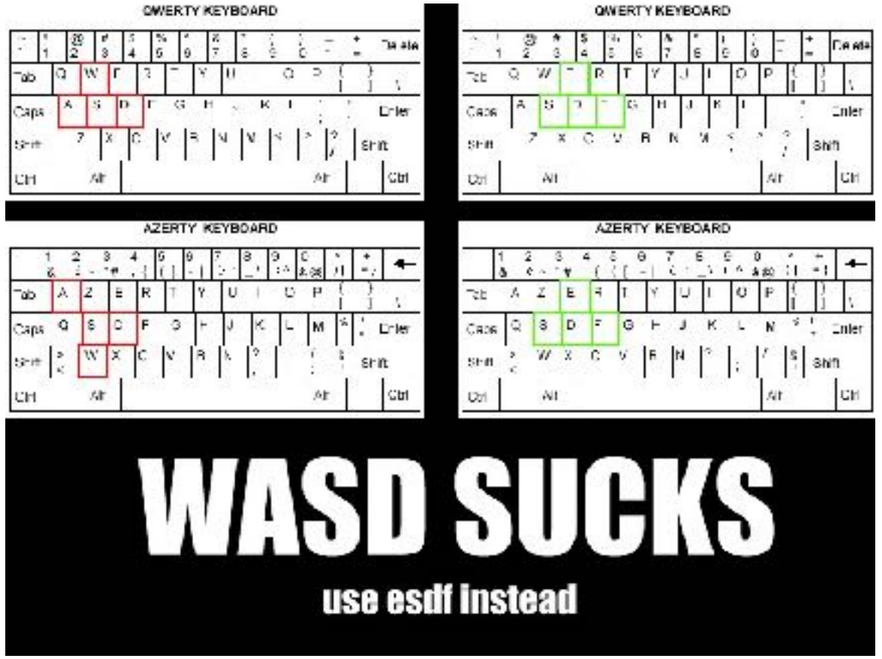
5 If the scan code is for one of the ordinary shift keys or for one of the special shift keys and toggle keys—Ctrl, Alt, Num Lock, Caps Lock, Scroll Lock, or Insert—the BIOS changes two bytes in a special area of memory to maintain a record of which of these keys has been pressed.

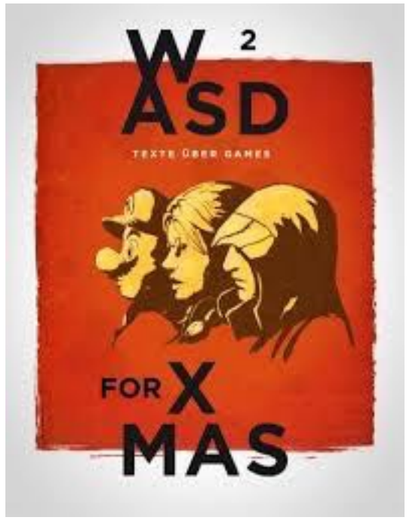
Buffer

6 For all other keys, the BIOS checks those two bytes to determine the status of the shift and toggle keys. Depending on the status indicated by those bytes, the BIOS translates the appropriate scan code into an ASCII code, used by the PC, that stands for a character, or into a special code for a function key or a cursor movement key. Uppercase and lowercase characters have different ASCII codes. Applications can choose to interpret any keystroke to display a character, or as a command. Ctrl+B, for example, is universally used by Windows applications to toggle the boldface attribute. In either case, the BIOS places the ASCII or special key code into its own memory buffer, where it is retrieved by the operating system or application software as soon as any current operation is finished.

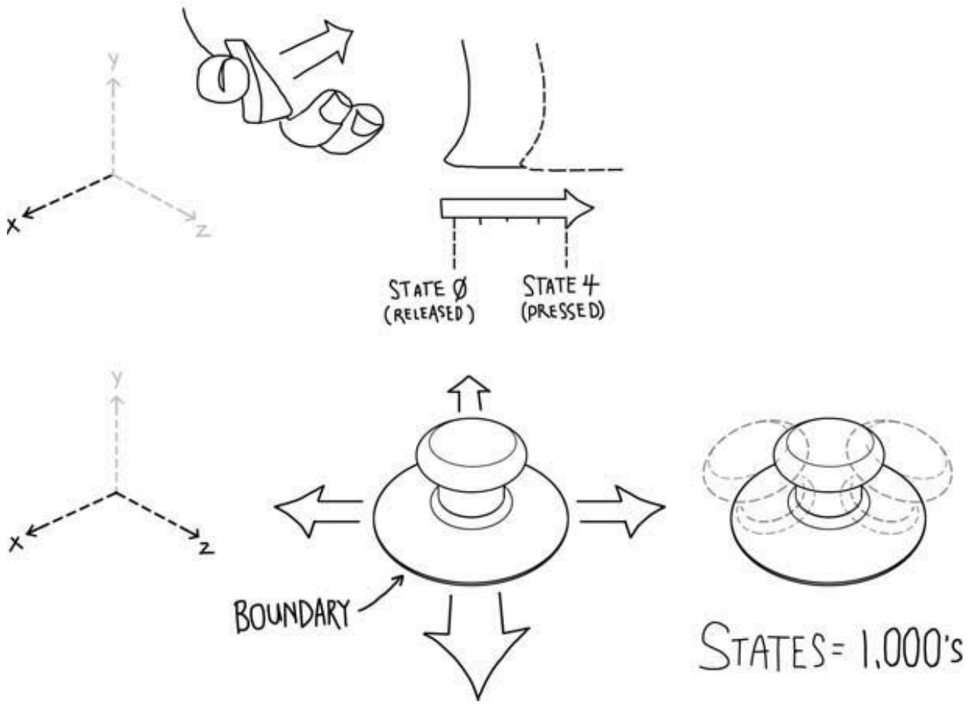
The Typing of the Dead

Keyboard Layout





Klassifikation von Eingabegeräten



nach Jacob (1996): The Future of Input Devices

Hand – Kontinuierlich

Bewegungsart: Linear vs. Rotation

Sensoring: Position, Bewegung, Kraft

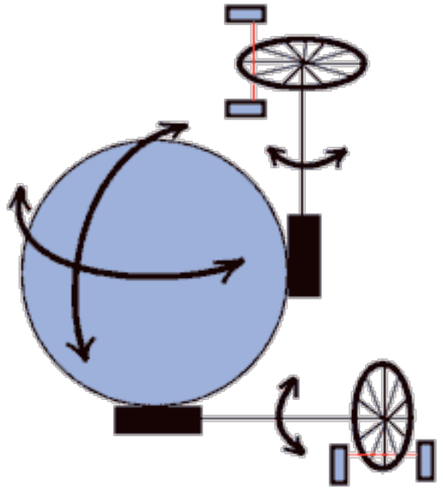
Bewegungs-Dimensionen: 1–3

Direkte vs. Indirekte Eingabe: z.B: Maus vs. Touchscreen

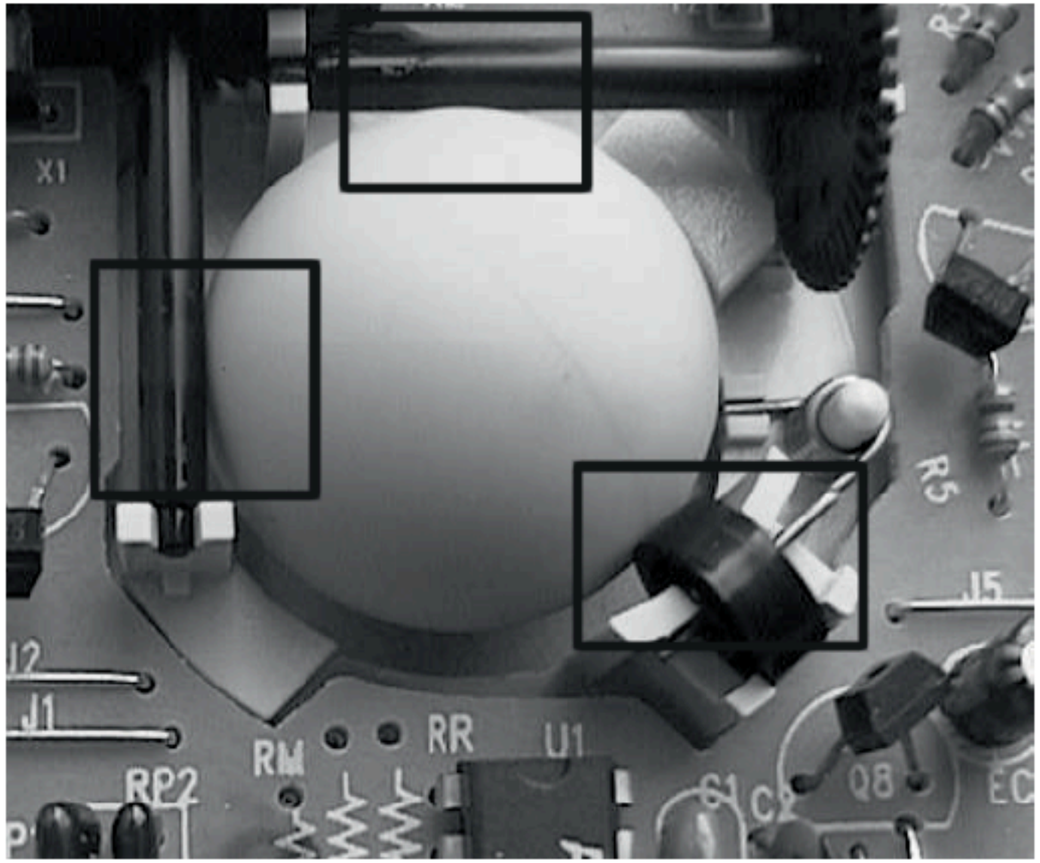
Grenzen der Bewegung

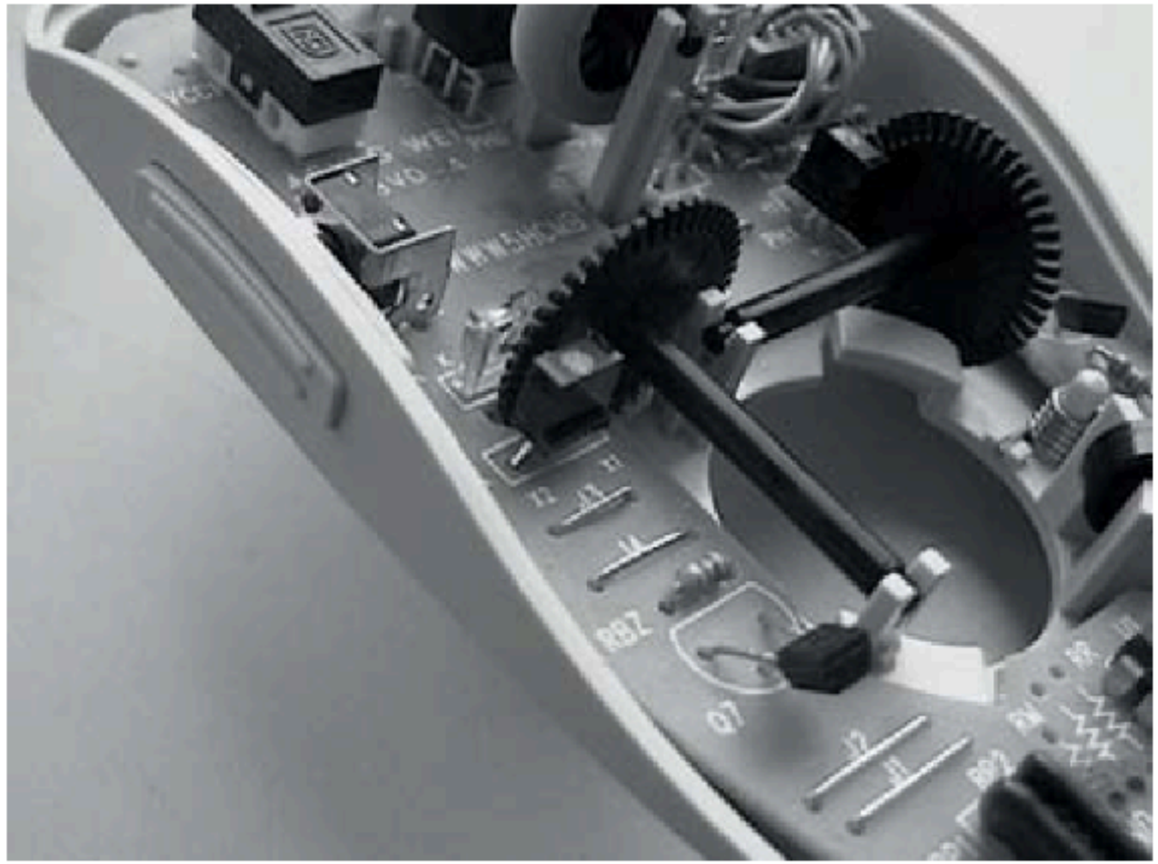
Empfindlichkeit: Anzahl der Messwerte

Signale: Datentypen

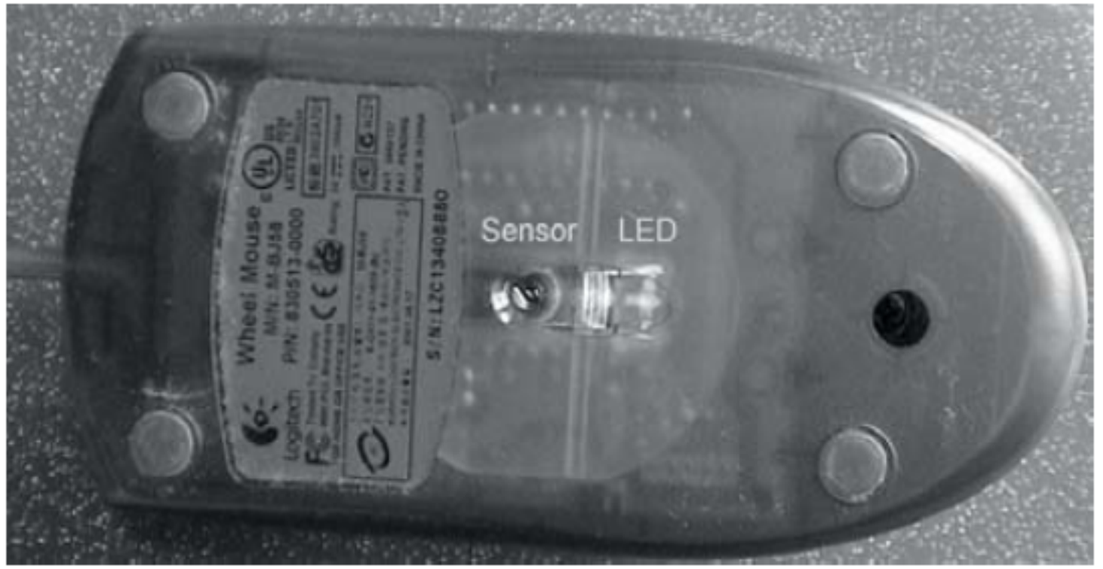


Mechanische Maus

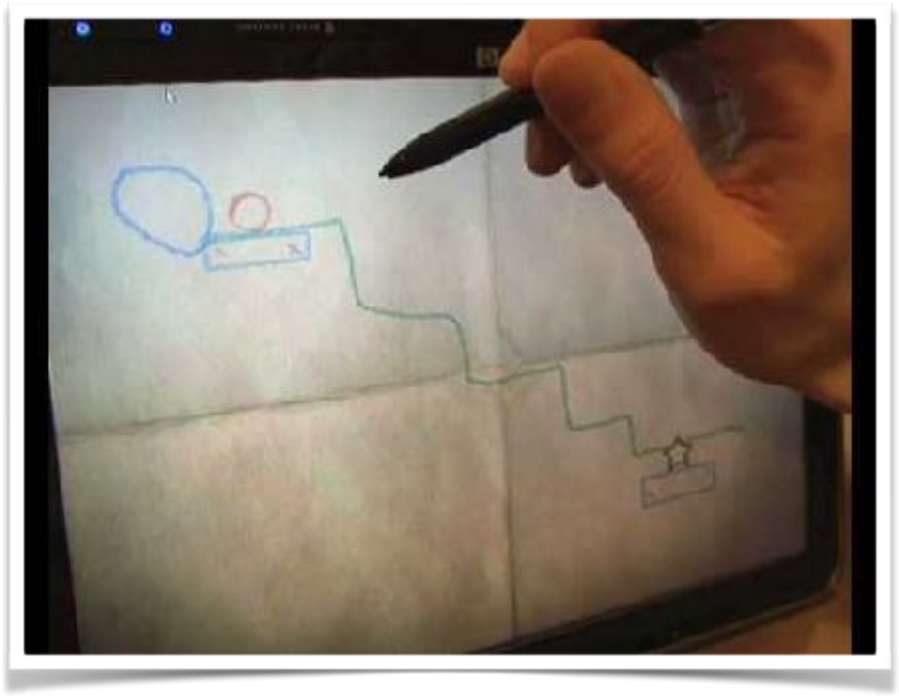




Optische Maus



Touchscreen Controls



Crayon Physics Deluxe

Fingle

Touch-Screen

Jetpack Joyride
One Button Game

Being able to play with one hand only is essential

Die Hand kann Teile des Bildschirms verdecken

https://www.deconstructoroffun.com/blog//2013/02/monetizing-infinite-runner.html

Spezielle Controller

http://www.grunge.com/12279/gaming-accessories-way-ahead-time/


2014
NEBUTAYA
HAKUKOU TONDSAKI
SHUNICHI KITAMURA
TSUYOSHI TATSUTA
NEBUTAYA.COM
LEFT BLOCK
CENTER BLOCK
TOGGLE SWITCHES
MAIN
SUB
CHAFF
MISSILE CHANGE
MAIN OPERATION
MAIN OPERATION SWITCH

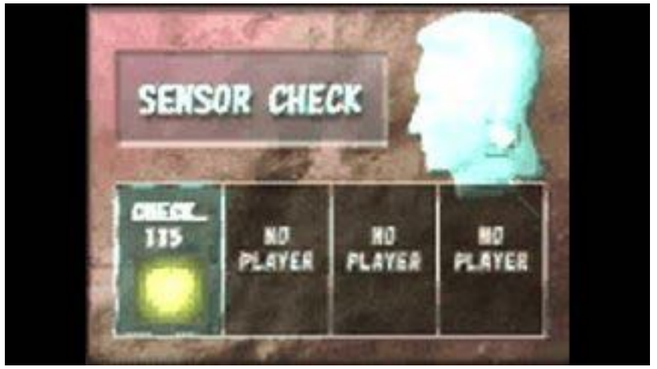
https://www.polygon.com/2016/7/16/12207114/evo-2016-player-piano-keyboard-controller

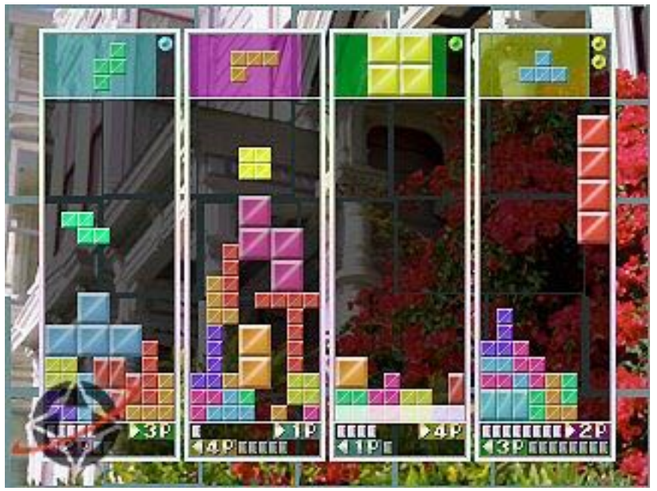
Sammlerstücke

Pip-Boy Edition von Fallout 4


NOFX

Biofeedback – Puls

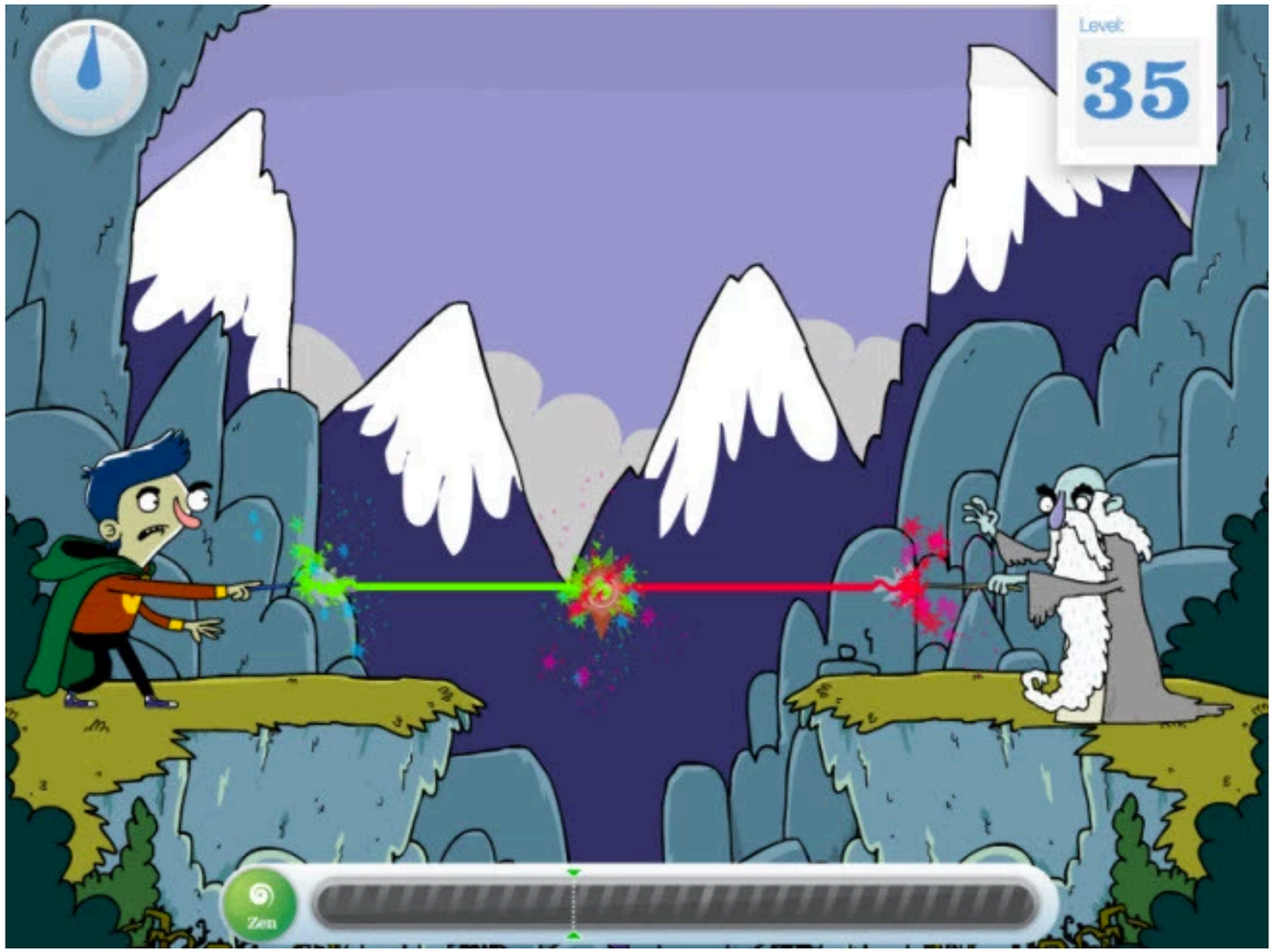




Tetris 64

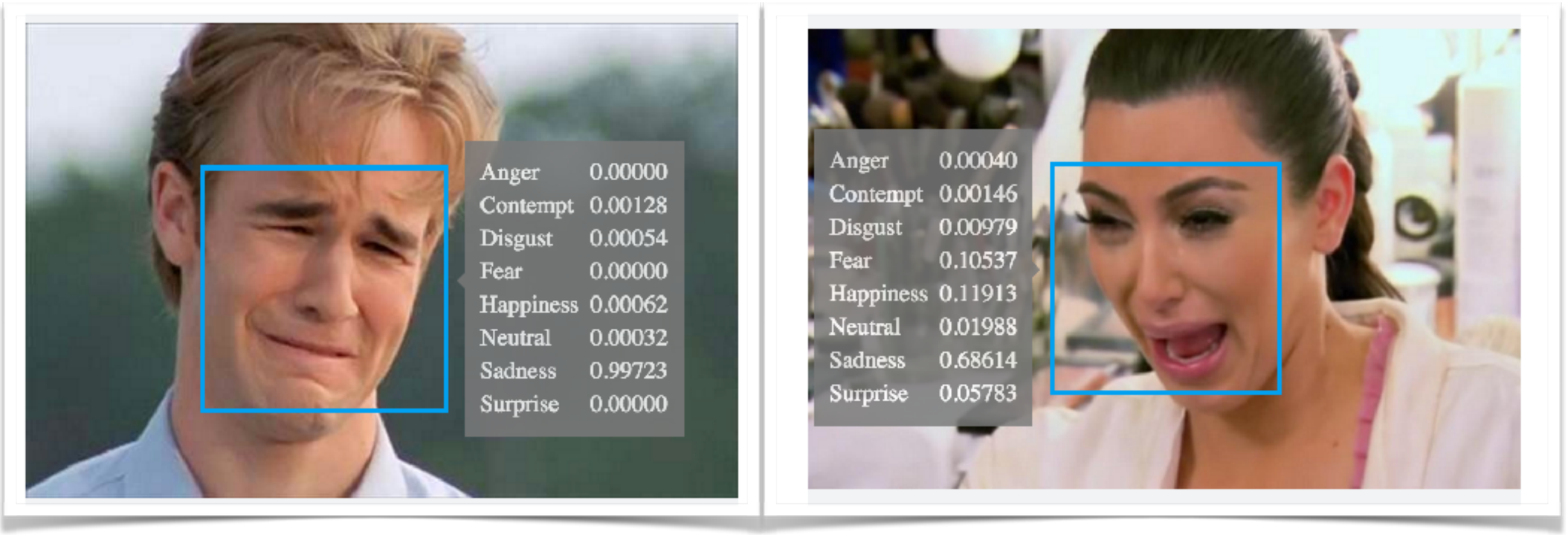
Biofeedback – Nevermind

Gehirnwellen



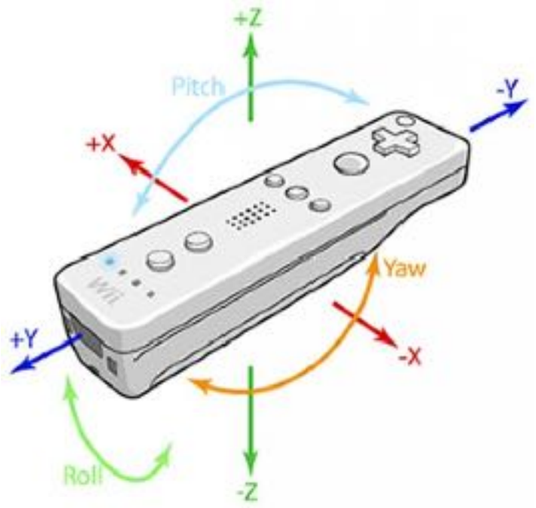
NeuroSky: Focus Pocus

Gesichtserkennung

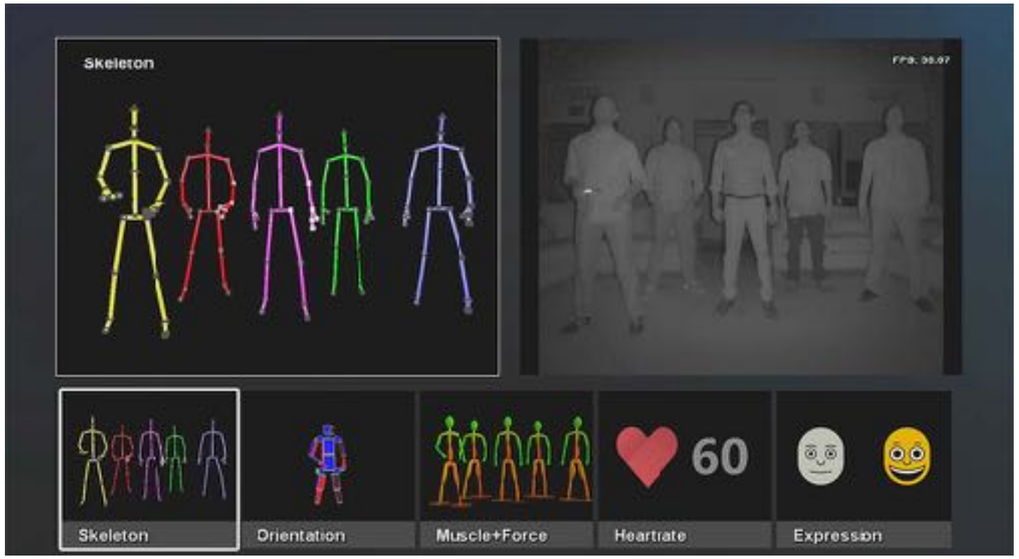


Project Oxford
https://www.microsoft.com/cognitive-services/

Embodied Interface



Wii Mote

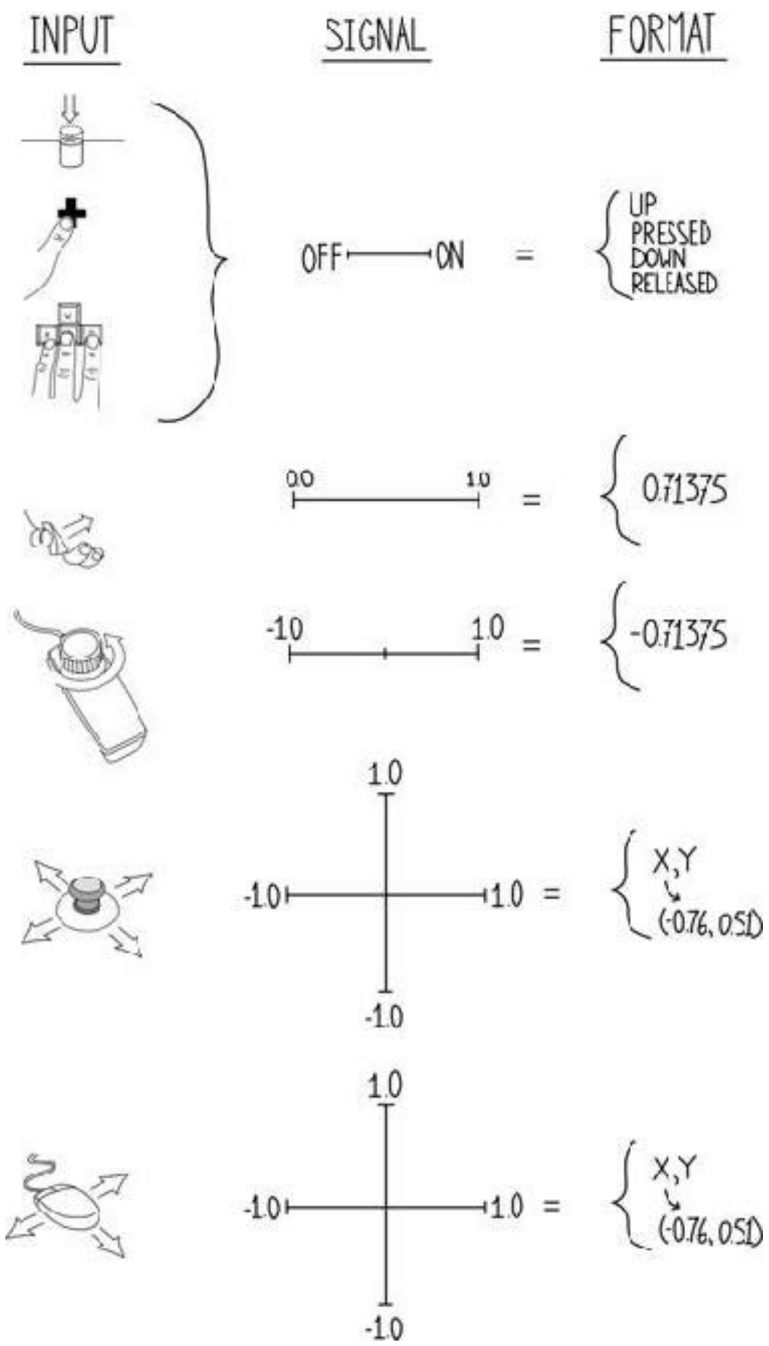


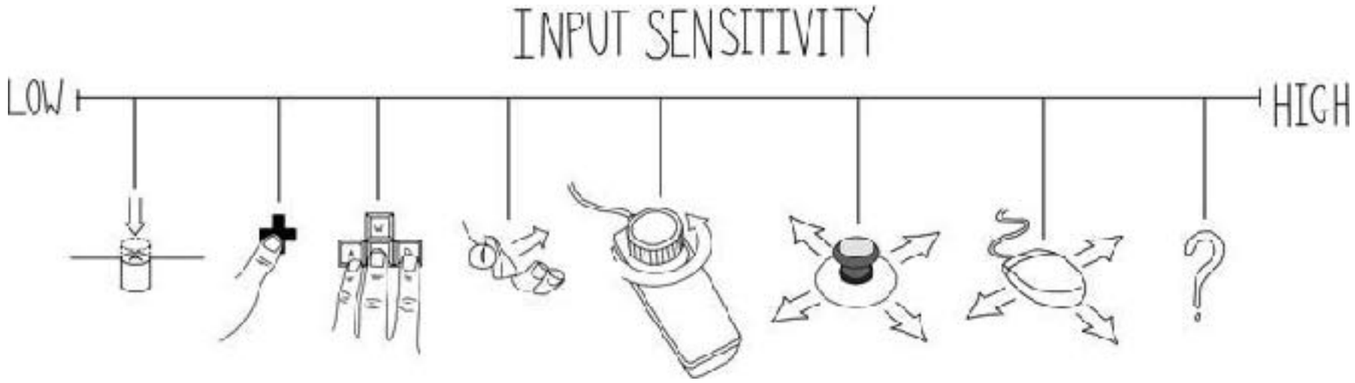
MS Kinect 2

PS Move

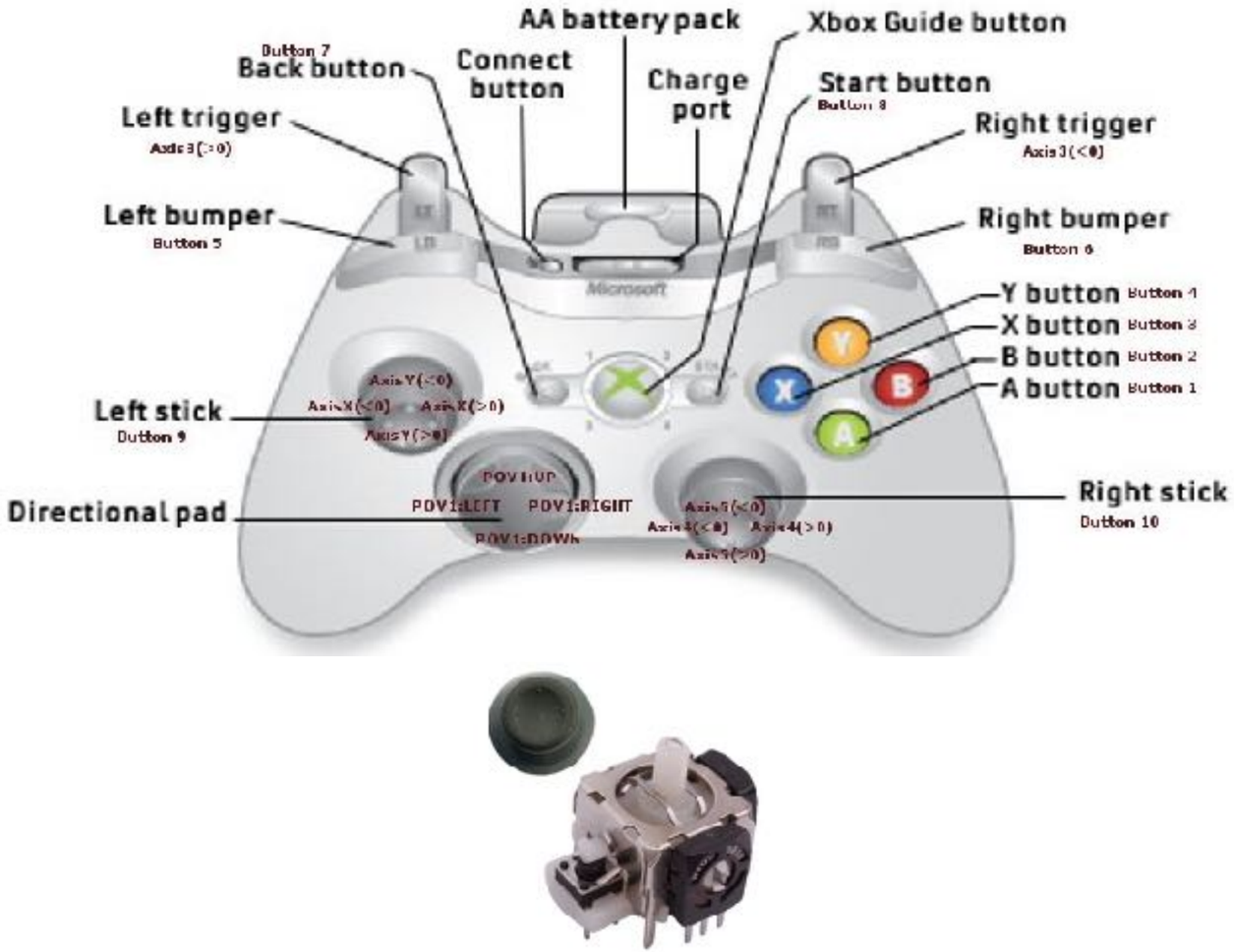
- PS Eye
- WiiMote
- PS Move
- MS Kinect
- Leap Motion
- HTC Vive
- Oculus Touch

Signale


INPUT
SIGNAL
FORMAT
OFF
ON
=
UP
PRESSED
DOWN
RELEASED
00
1.0
=
0.71375
-1.0
1.0
=
-0.71375
1.0
-1.0
1.0
=
X,Y
(-0.76, 0.51)
-1.0
1.0
-1.0
1.0
=
X,Y
(-0.76, 0.51)
-1.0


INPUT SENSITIVITY
LOW
HIGH
?

Xbox360-Controller



https://katyscode.wordpress.com/2013/08/30/xinput-tutorial-part-1-adding-gamepad-support-to-your-windows-game/

Face buttons
XINPUT_GAMEPAD_A
XINPUT_GAMEPAD_B
XINPUT_GAMEPAD_X
XINPUT_GAMEPAD_Y

Directional pad arrows
XINPUT_GAMEPAD_DPAD_LEFT
XINPUT_GAMEPAD_DPAD_RIGHT
XINPUT_GAMEPAD_DPAD_UP
XINPUT_GAMEPAD_DPAD_DOWN

Shoulder buttons
XINPUT_GAMEPAD_LEFT_SHOULDER
XINPUT_GAMEPAD_RIGHT_SHOULDER

Analog thumb sticks (when pressed in and used as a button)
XINPUT_GAMEPAD_LEFT_THUMB
XINPUT_GAMEPAD_RIGHT_THUMB

Centre buttons
XINPUT_GAMEPAD_BACK
XINPUT_GAMEPAD_START

```
bool A_button_pressed = ((state.Gamepad.wButtons & XINPUT_GAMEPAD_A) != 0);
```

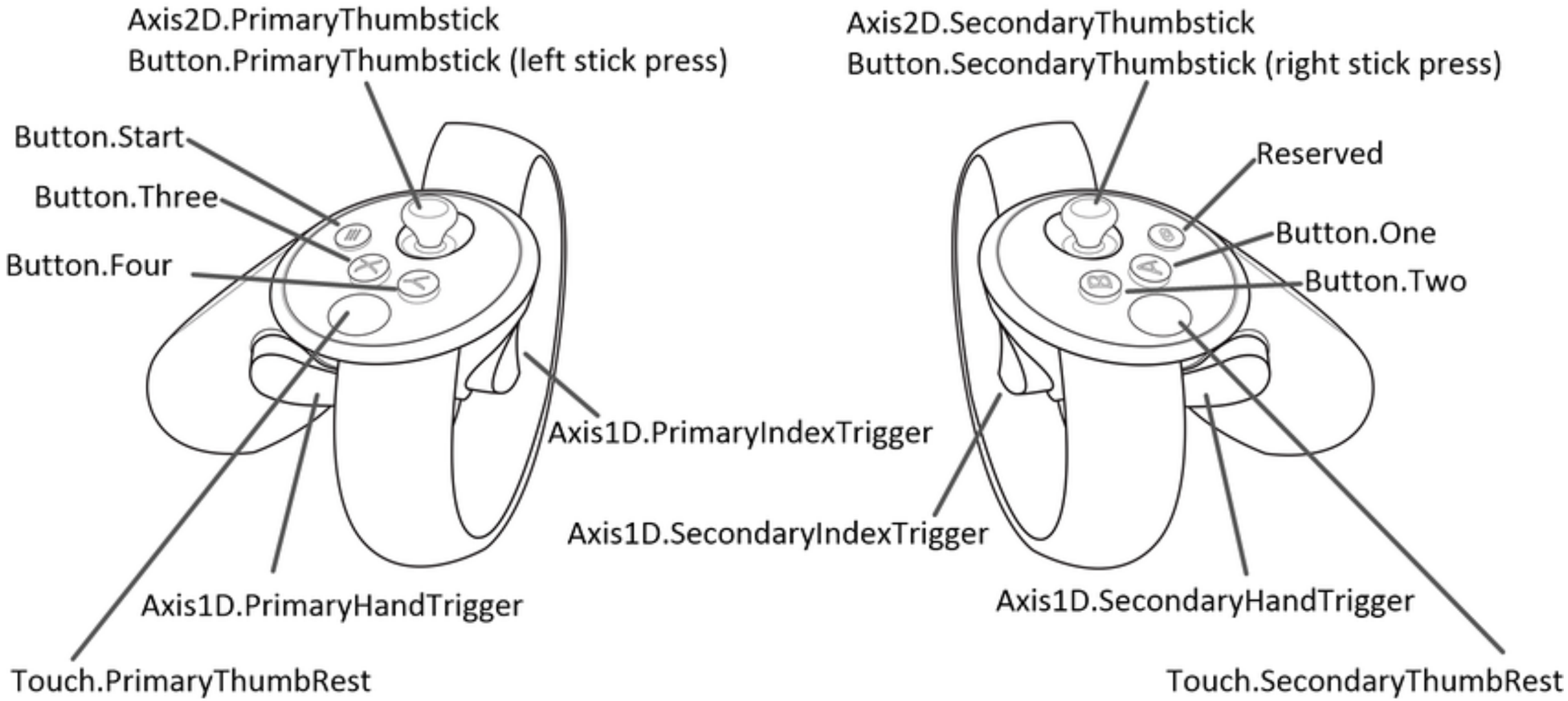
Trigger
The values returned by the triggers are unsigned 8-bit integers (range 0-255 where 0 is not pressed and 255 is fully pressed)

```
float leftTrigger = (float) state.Gamepad.bLeftTrigger / 255;
float rightTrigger = (float) state.Gamepad.bRightTrigger / 255;
```

Analog Thumb Sticks
The values returned by the thumb sticks are signed 16-bit integers (range -32768 to +32767 where -32768 is fully to the left or down, 0 is centered and 32767 is fully to the right or up, depending on the axis being queried)

```
float normLX = fmaxf(-1, (float) state.Gamepad.sThumbLX / 32767);
float normLY = fmaxf(-1, (float) state.Gamepad.sThumbLY / 32767);
```

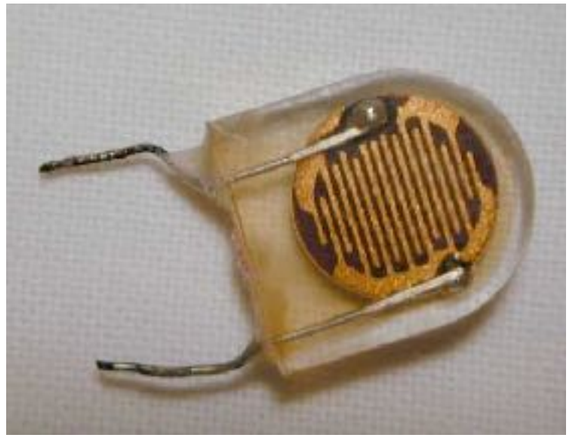
Oculus Touch



https://docs.unity3d.com/Manual/OculusControllers.html

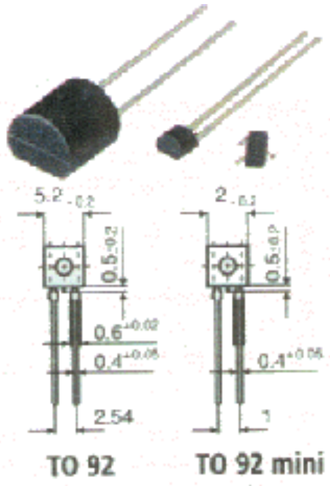
Sensoren

Biegesensor

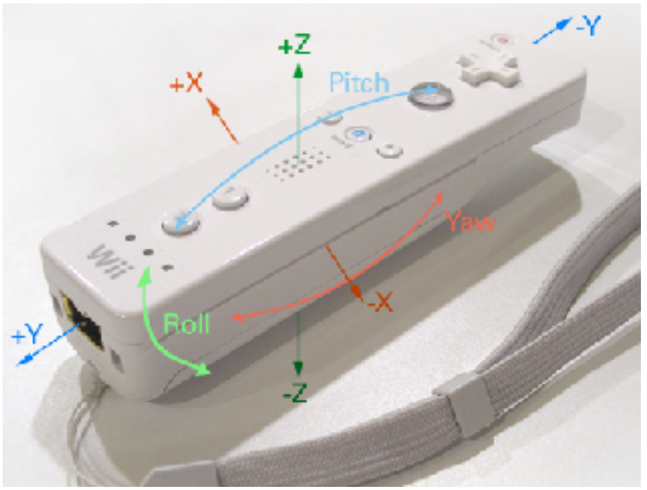


Lichtsensor

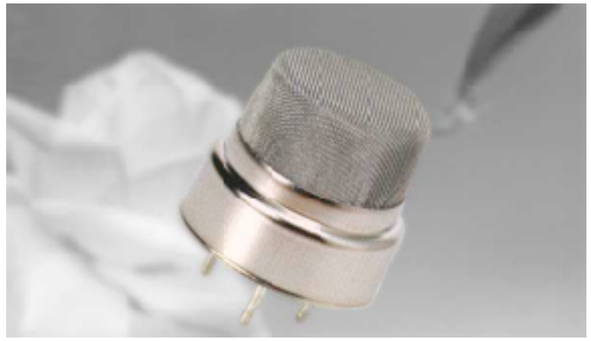
Schwerkraftsensor



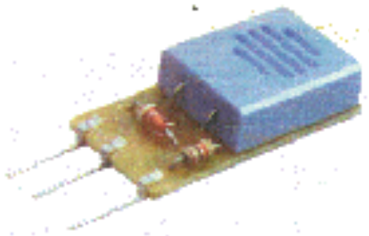
Temperatursensoren



Beschleunigungssensoren

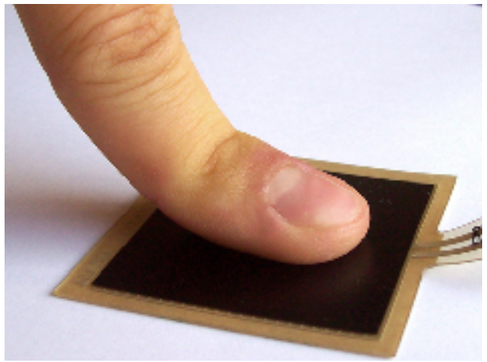


Gassensor



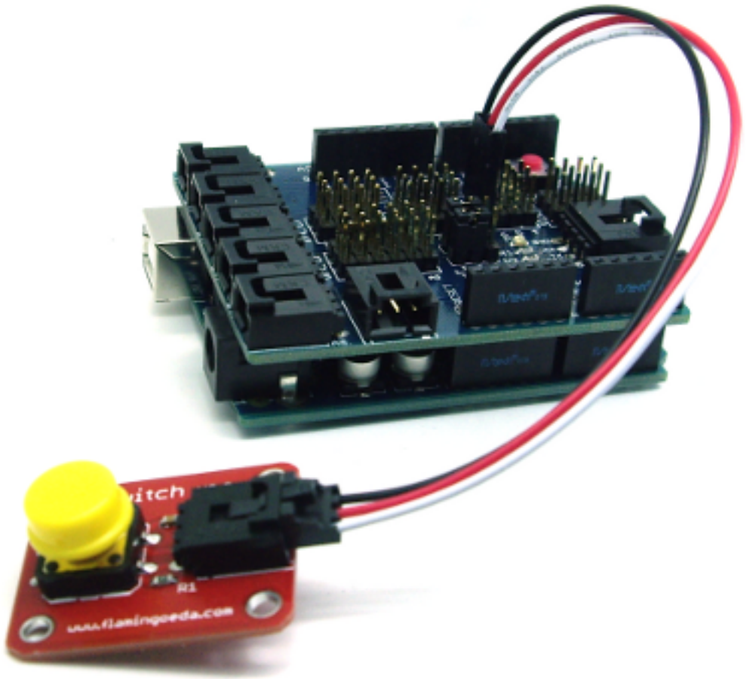
Feuchtigkeitssensor

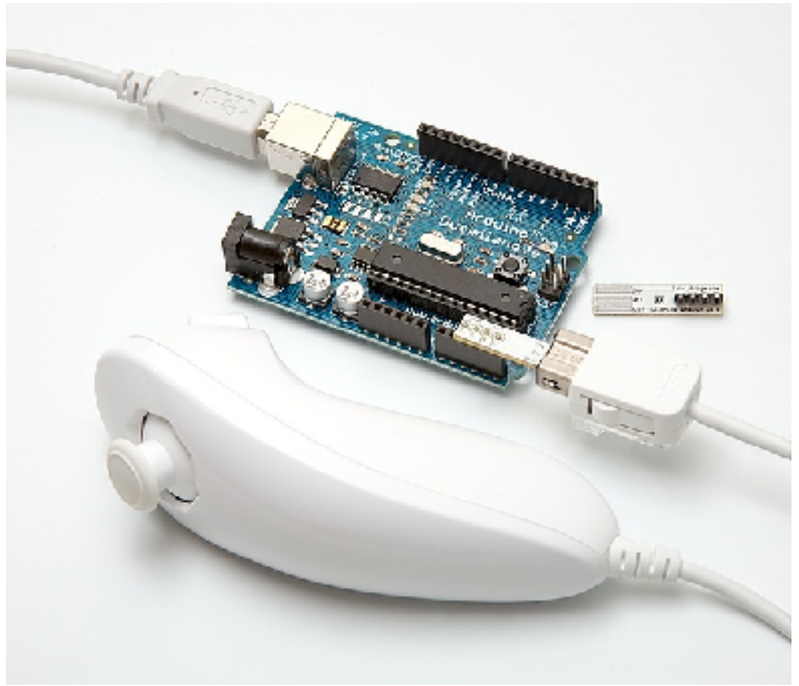
Schallsensor/Mikrofon



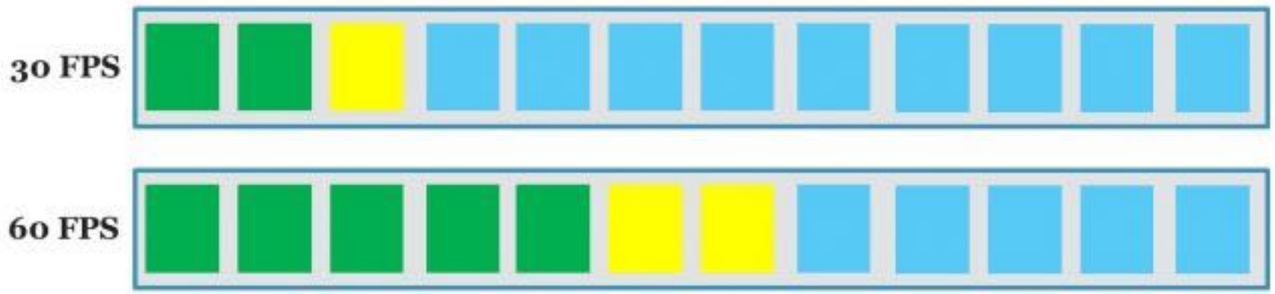
Drucksensor

Arduino





Technische Verarbeitung



For human perception window of ~80-100 ms,
maximum comfortable lag on the screen is around:
2-3 frames max for 30 FPS.
5-7 frames max for 60 FPS.

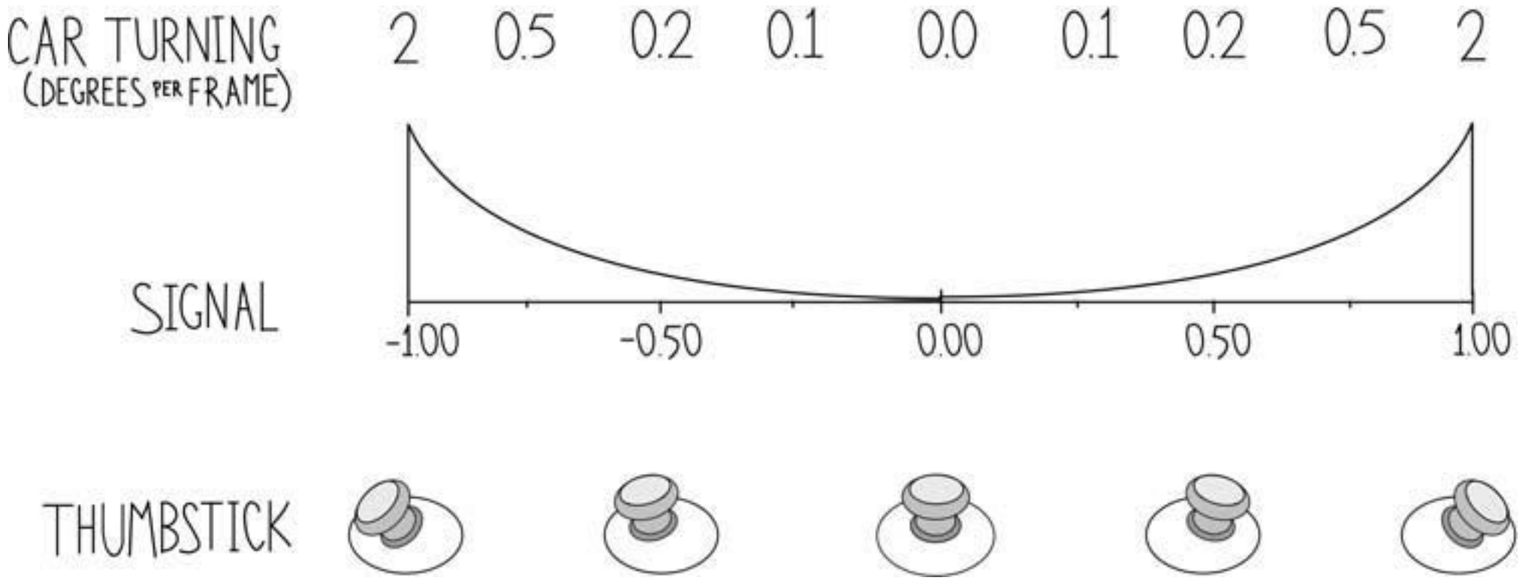
There's always 1-2 frames delay from the hardware.

Complex signal filtering adds delay.

Long animations and non-interruptible actions might not fit into
100 ms< perception window and create the feeling of "lag."

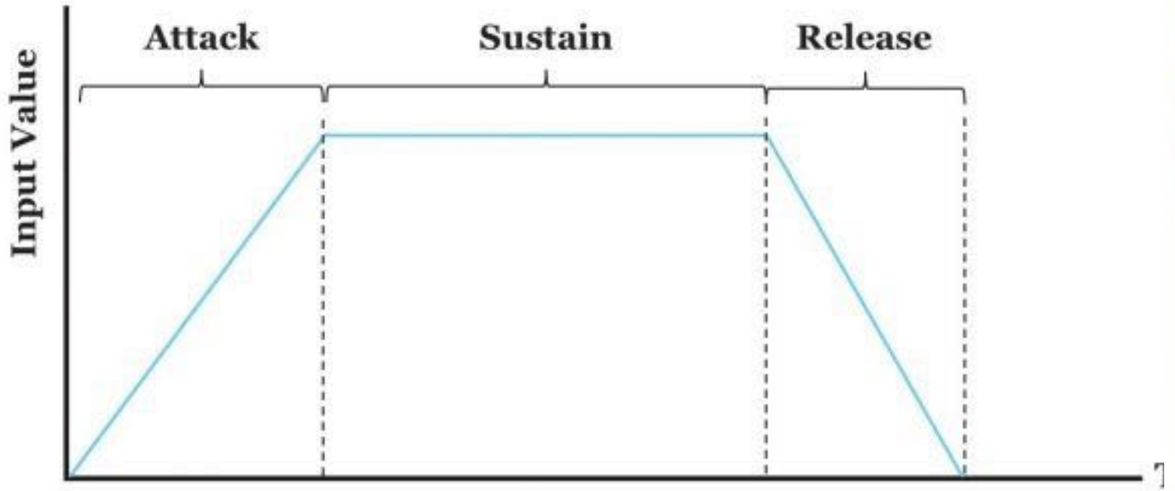
V-Sync adds 1-2 frames delay.

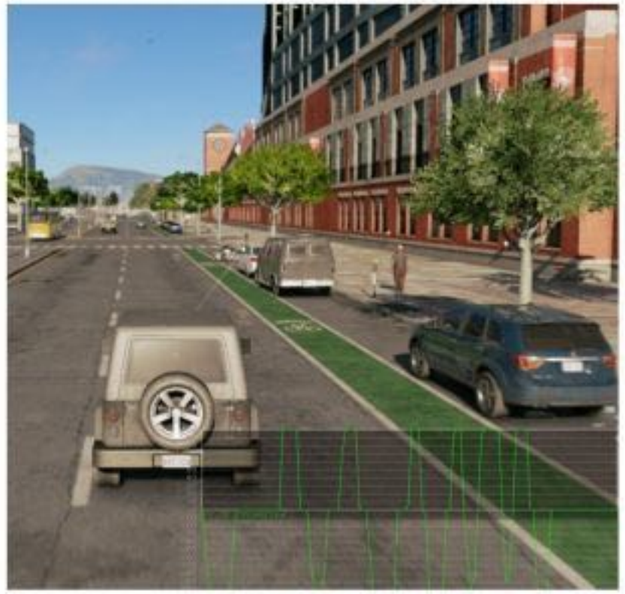
Filter


CAR TURNING
(DEGREES PER FRAME)
2 0.5 0.2 0.1 0.0 0.1 0.2 0.5 2
SIGNAL
-1.00 -0.50 0.00 0.50 1.00
THUMBSTICK

Filter – ADSR

Curves – filter signal strength depends on time/speed.





On the left, you can see the gamepad input curve. To mimic this signal for the keyboard, we used an input curve that takes into account how long the steering button was pressed, which is allowed to turn the digital input from the keyboard into an analog signal. The curve is quite fast, with a very short delay time (the game has pretty arcade-style driving), but even such subtle mechanism makes control of the car much more natural and smoother.

http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

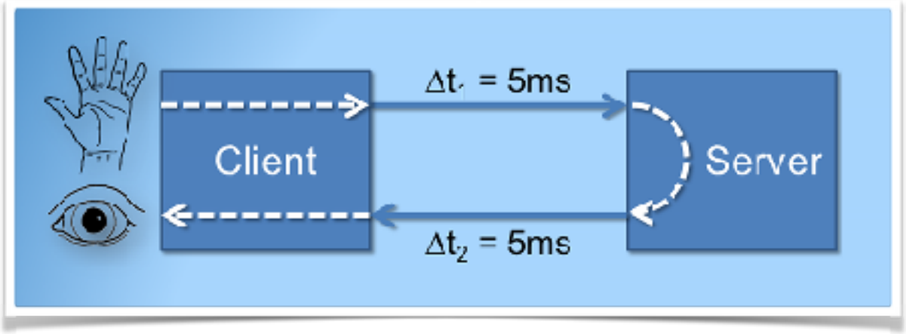
Response Time

http://cowboyprogramming.com/2008/05/30/measuring-responsiveness-in-video-games/

Note on the first frame (-1) , the finger is still moving, we start counting on the next frame (0) when the finger is fully on the button and the button is fully depressed. We then count until the first response, which comes at frame 12. This indicates a response time of 12/60ths. Since it's measured on the plama TV, we adjust this to 10/60ths. This gives us a raw response time for GTA-IV of 166 ms (200 ms on flat panel TVs).

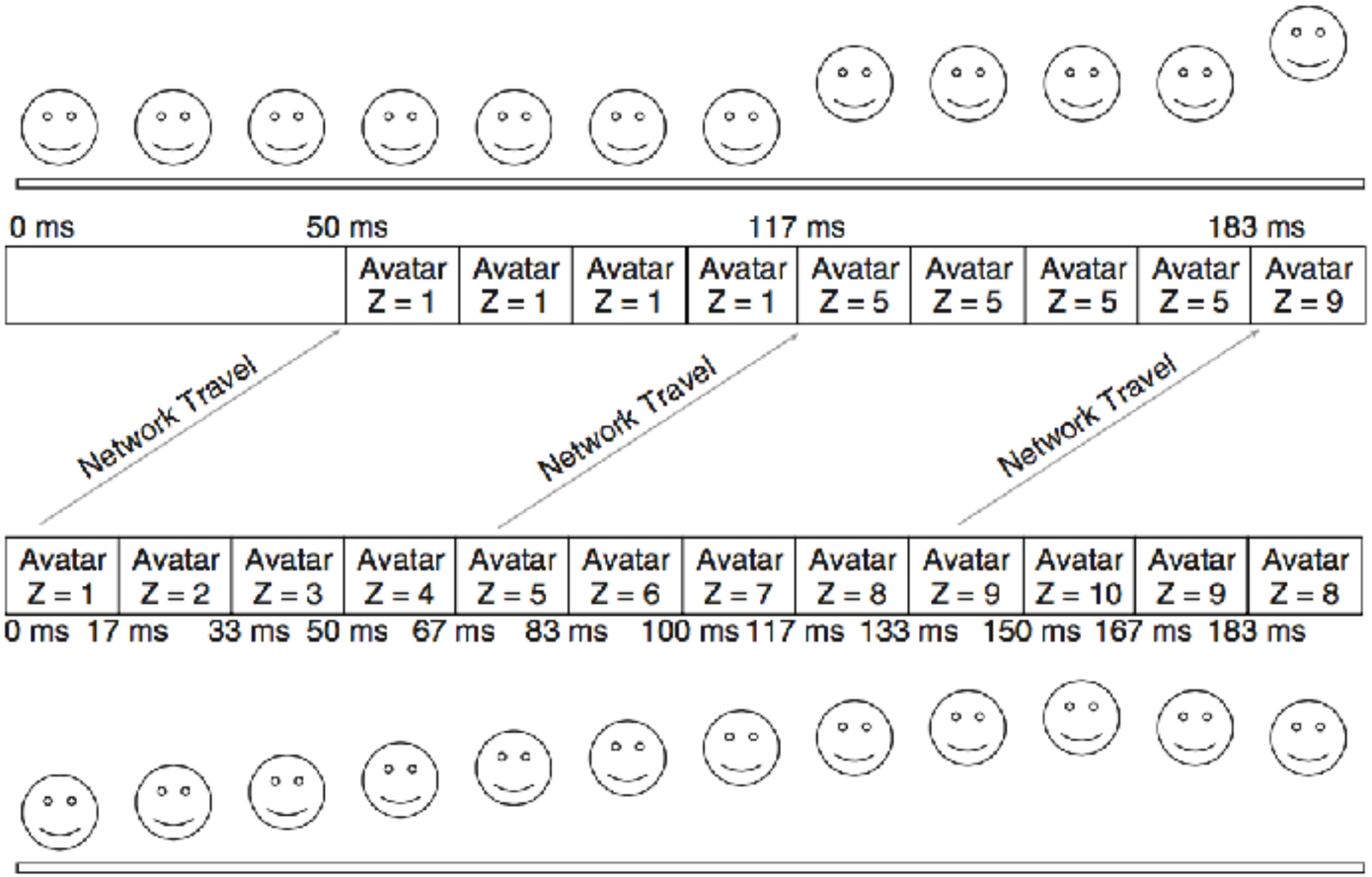
Latenz / Ping

Round Trip Time (RTT) ≈ 2 x Latenz



- Prozessorzeit
- Übertragungszeit
- Warteschlange
- Laufzeit:
  0.3 m/ns = 300 m / µs = 300 km / ms
  12.000 km benötigen 40 ms = 80 ms RTT

Lag


0 ms
50 ms
117 ms
183 ms
Avatar Z = 1
Avatar Z = 1
Avatar Z = 1
Avatar Z = 1
Avatar Z = 5
Avatar Z = 5
Avatar Z = 5
Avatar Z = 5
Avatar Z = 9
Network Travel
Network Travel
Network Travel
Avatar Z = 1
Avatar Z = 2
Avatar Z = 3
Avatar Z = 4
Avatar Z = 5
Avatar Z = 6
Avatar Z = 7
Avatar Z = 8
Avatar Z = 9
Avatar Z = 10
Avatar Z = 9
Avatar Z = 8
0 ms 17 ms 33 ms 50 ms 67 ms 83 ms 100 ms 117 ms 133 ms 150 ms 167 ms 183 ms



Interaktionsdesign

Affordance

affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used. .. Affordances provide strong clues to the operation of things. (Norman: The Psychology of Everyday Things, S. 9)

There are two main ways to use affordances in games:

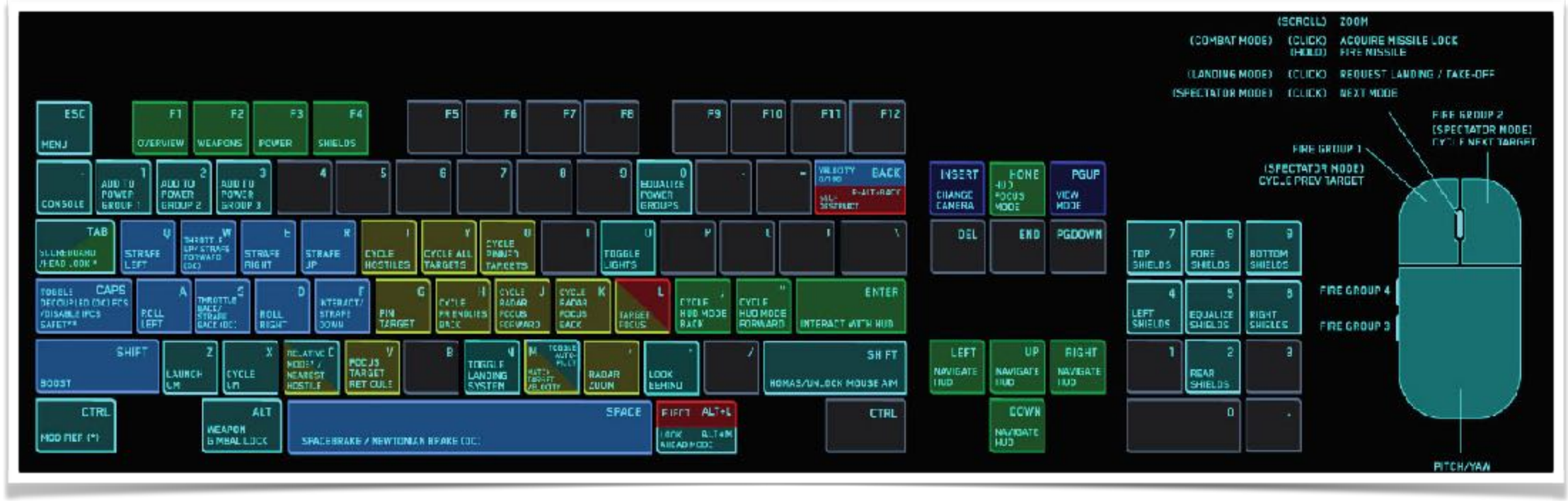
HUD – external UI elements that communicate what can be done with the object (control reminders, crosshair state, etc.). External HUD can even be a part of the game's narrative (Animus in Assassin's Creed, for example).

Game World – using of "Form Follows Function" principle when the form of the game object communicates what can be done with or by this object. It can be level design ("climbing" points on the building, barrels with "explosives" symbol, etc.), or the character animations/items (the smaller character is more agile, the bigger weapon is slower, etc.).

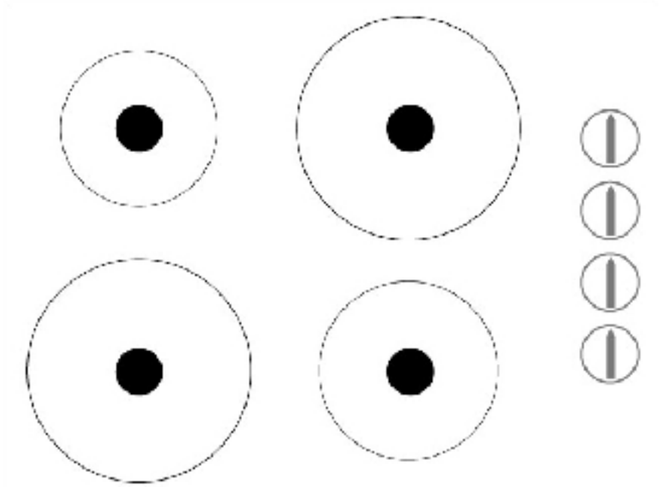
http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

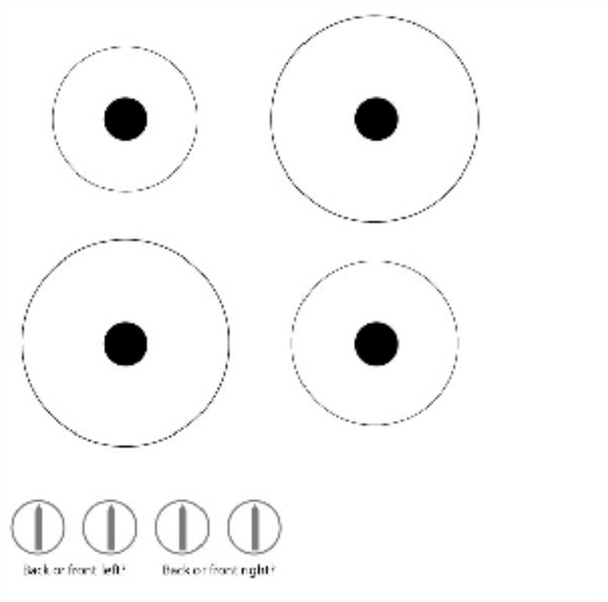
Sichtbarkeit (Visibility)

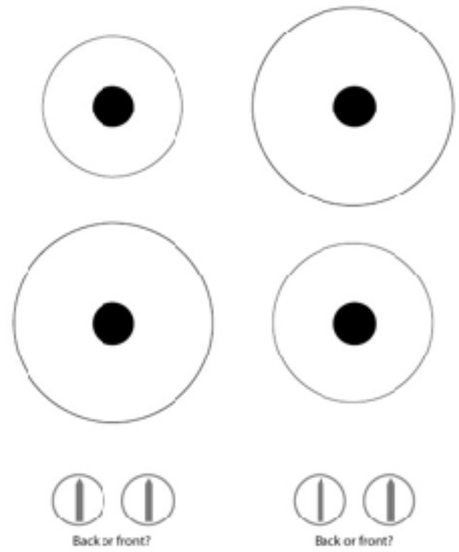
Visibility: the user needs help. Just the right things have to be visible: to indicate what parts operate and how, to indicate how the user is to interact with the device. Visibility indicates the mapping between intended actions and actual operations. (Norman, 8)

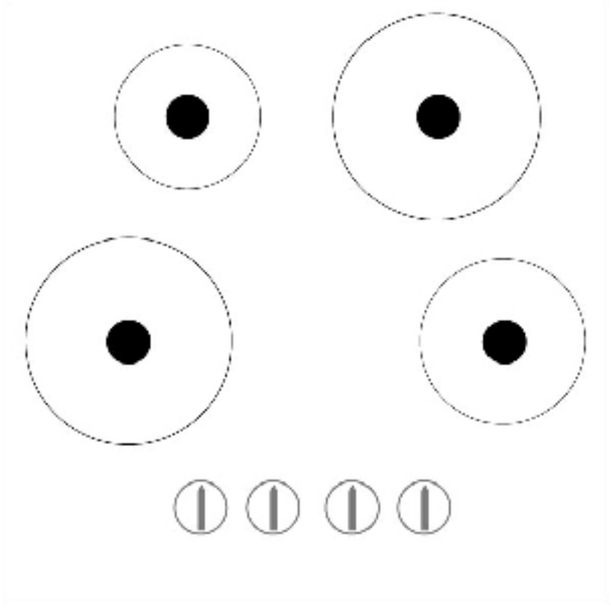


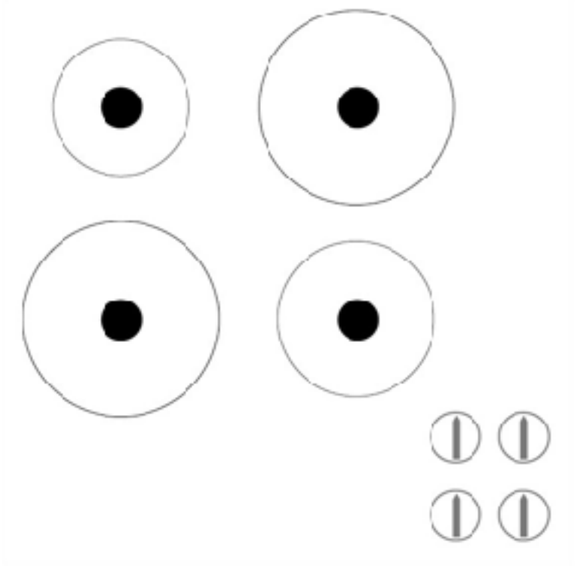
Star Citizen











Mapping

Natural mapping, by which I mean taking advantage of physical analogies and cultural standards, leads to immediate understanding.

Norman, S. 23

Mapping

https://www.youtube.com/watch?v=96uSY-YXbIM

Beziehung zwischen Spielerhandlung und Spielhandlung

Natural Mapping

Arbitrary Mapping

Exagerated Mapping

Control interruption

Physical Endurance

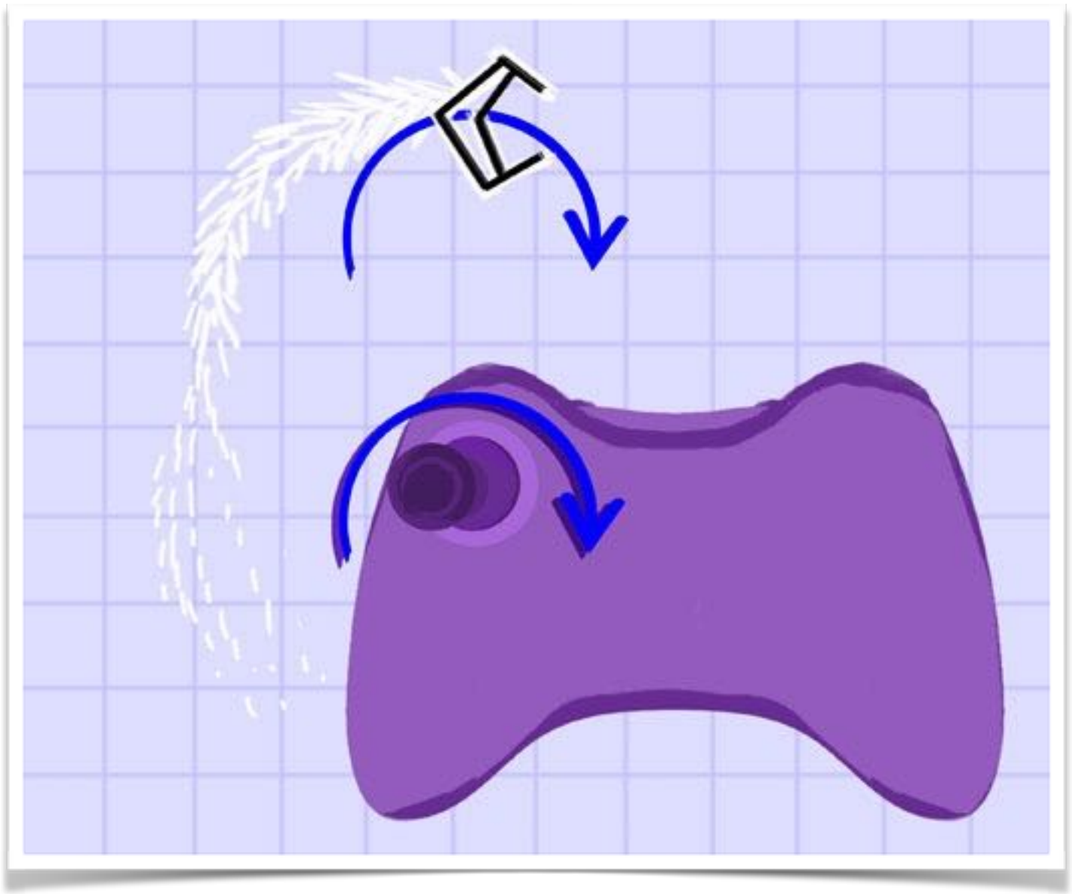
Tiptoeing

Stunning

Consistency

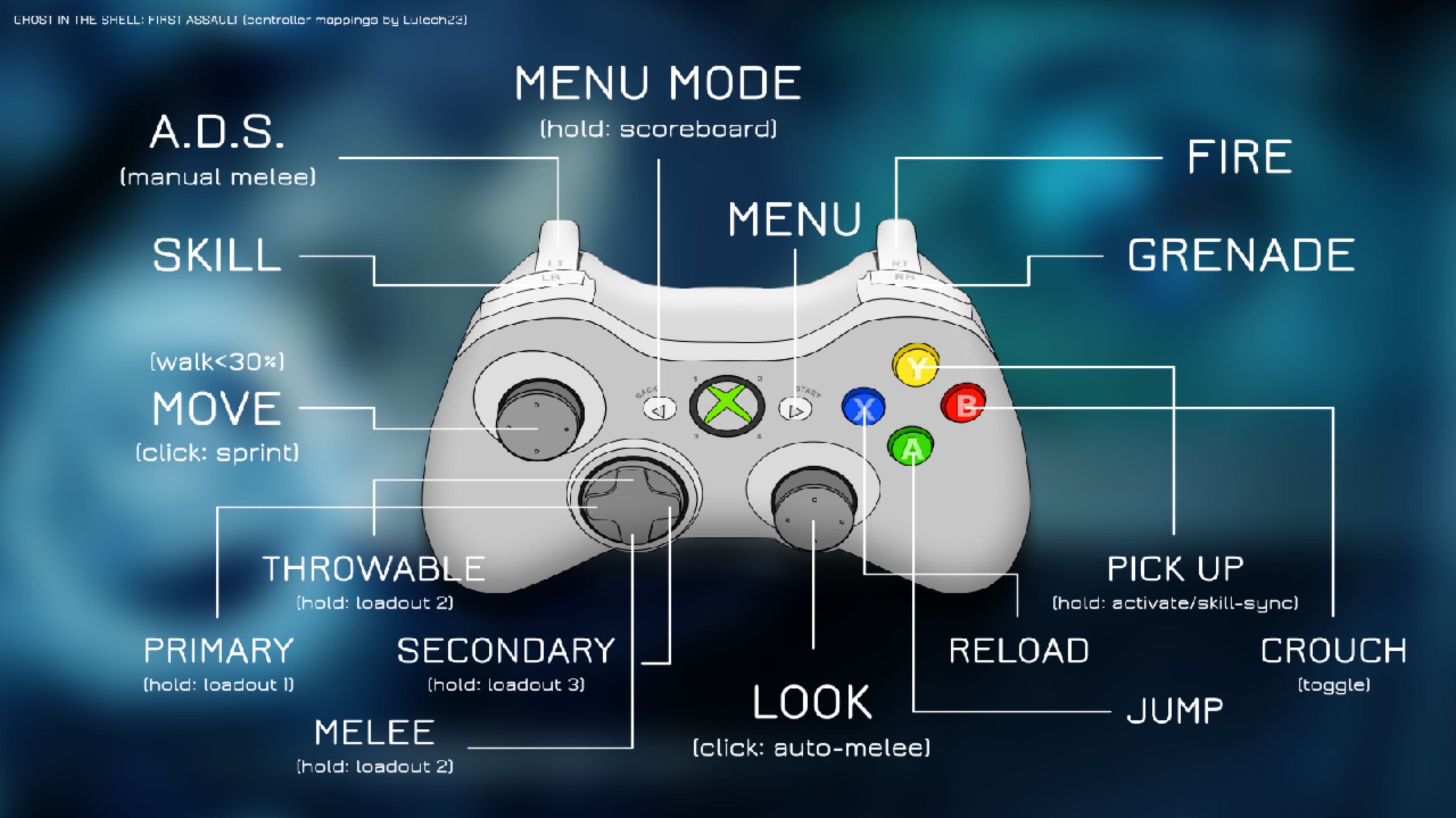
Gonzalo Frasca: Play the Message, S 152 ff.

Natural Mapping



Geometry Wars

GHOST IN THE SHELL: FIRST ASSAULT (controller mappings by Lulech23)

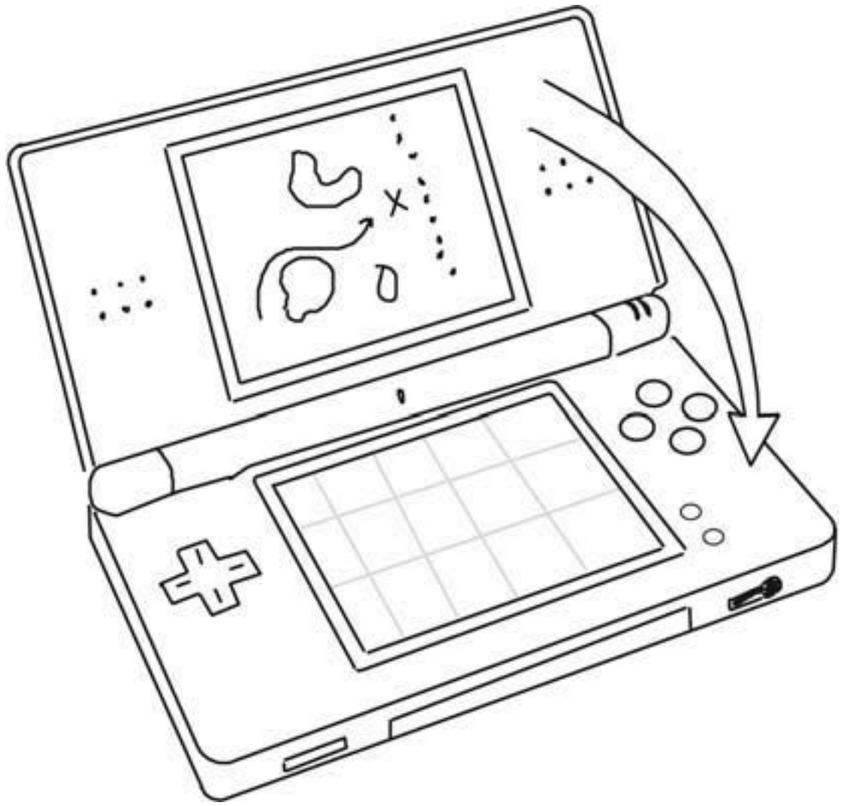


Ludo-Performative Dissonance

Ludo-Performative Consonance

Konsistenz

Die wichtigste Konsistenz ist Konsistenz mit Nutzererwartungen.



The Legend of Zelda: Phantom Hourglass players are told to step up to an altar and “stamp” their “map” with the location of a new area to be explored later

Swink: Game Feel, S. 55 f.

Feedback

s. 08 Sensory

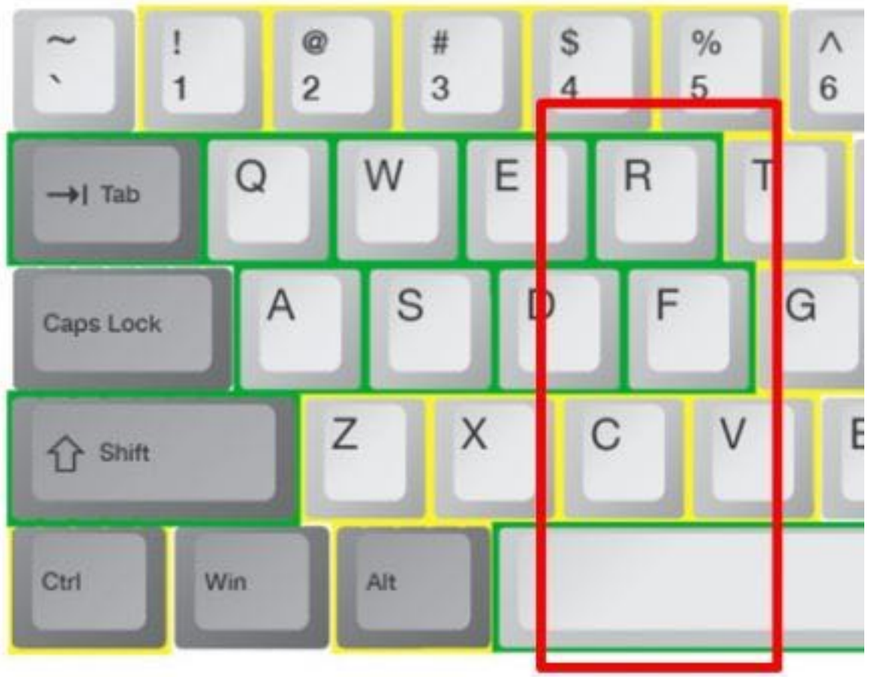
Accuracy: Response Time

Visual Feedback: Animation, Visual effects, HUD

Audio Feedback

Tactile Feedback

Konzeptuelle Modelle



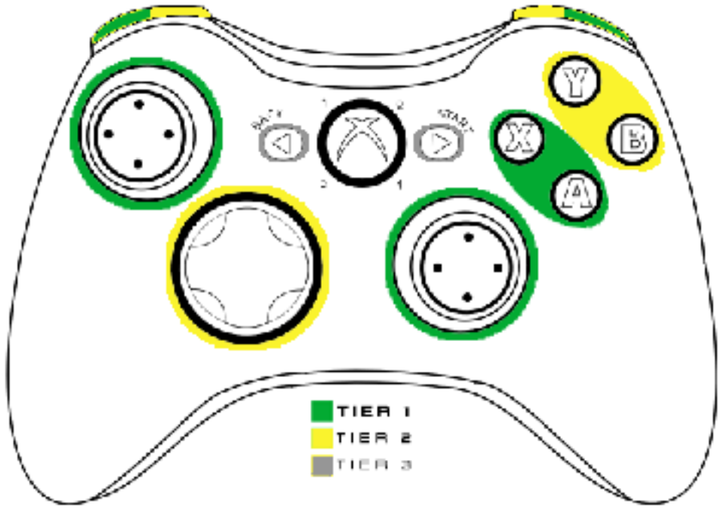
http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

Gruppieren

People learn and memorize by making patterns. To decrease memory load and improve learning, divide controls into logical groups:

- Similar actions should be in one group – all move actions are in one group, all combat actions in another group, etc. Grouped actions that are related to one basic mechanic (combat, driving, navigation, etc.) are much easier to move to automated state (WASD).
- Groups should take in account hand limitations – should match accessibility tiers.

Gruppieren



The most frequent actions should be in the most accessible places and match primary control group of the player's hand.

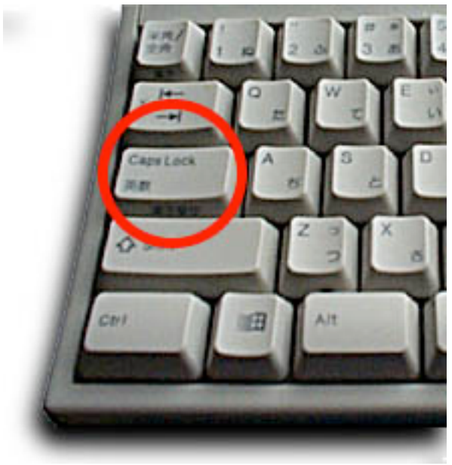
- Groups should be consistent, if you have more than one layout – similar actions in different layouts should work on the same button (ex: "Sprint" on [Shift] in On Foot layout and "Nitro" on [Shift] in Driving layout).
- Two biggest groups are the player's two hands – if you have two important actions (or groups of actions) that the player should use simultaneously, divide them between two hands, it will make memorization easier.

http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

Modale Interfaces

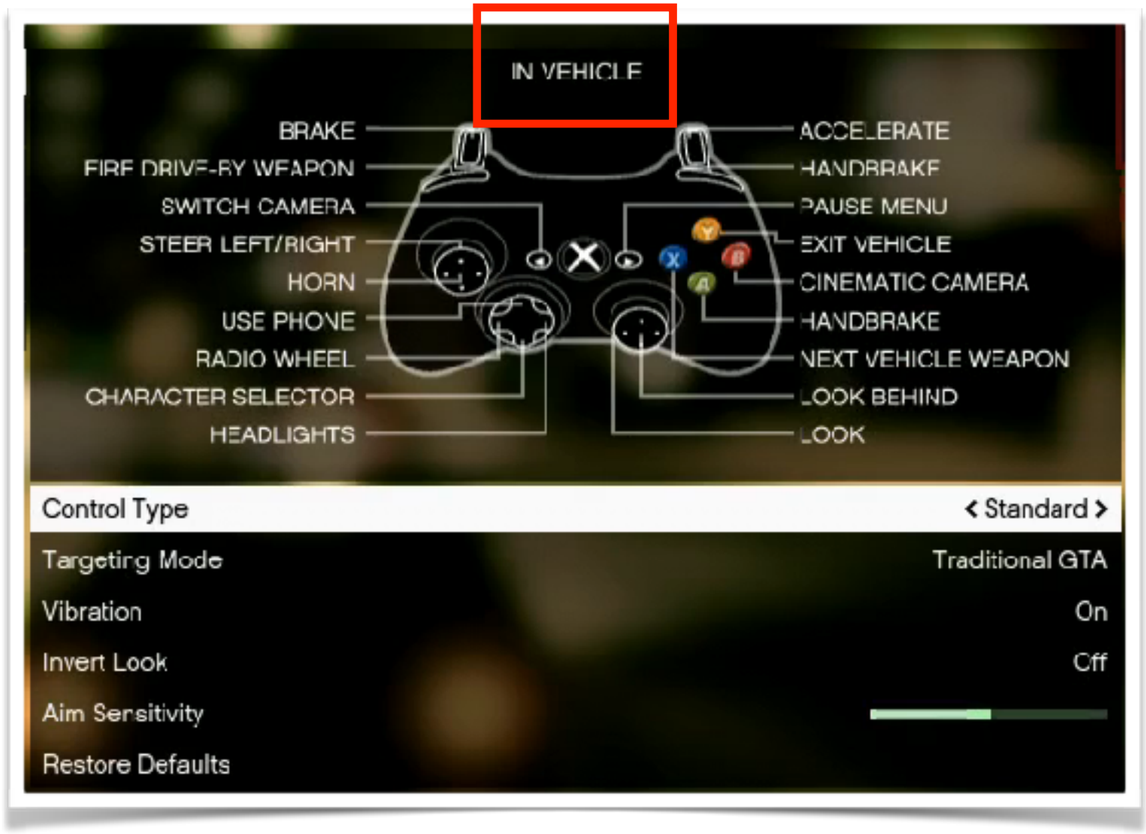
These switches on an RPV controller cannot be operated accidentally; the handle must be pulled out before the switch can be operated

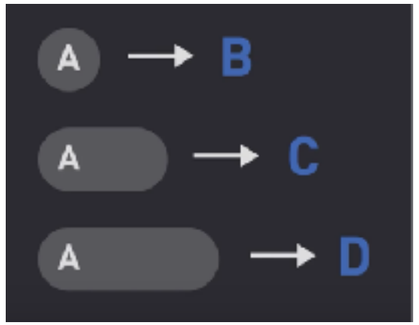
"A human-machine interface is modal with respect to a given gesture when (1) the current state of the interface is not the user's locus of attention and (2) the interface will execute one among several different responses to the gesture, depending on the system's current state." (Rafkin, Human Computer Interface, S. 42).

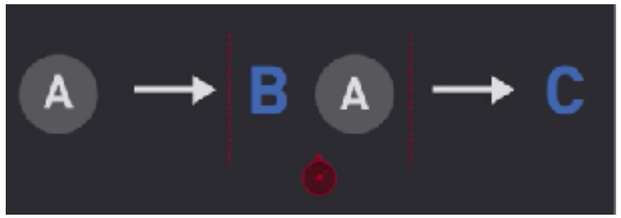


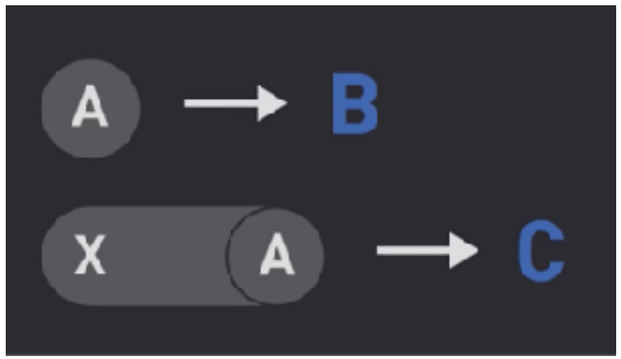
Zustände

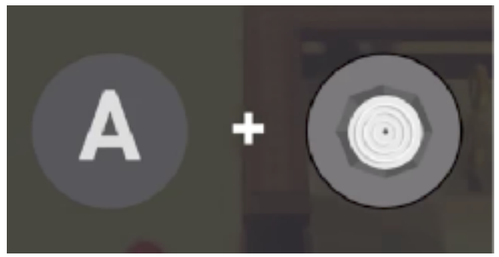
Kontextabhängige Controller-Belegung











Versatile Buttons (Verbs)

https://www.youtube.com/watch?v=7daTGyVZ60I

Duration (Charge Action)

Multi Press (Double Jump, QTE)

Combine (Combos)

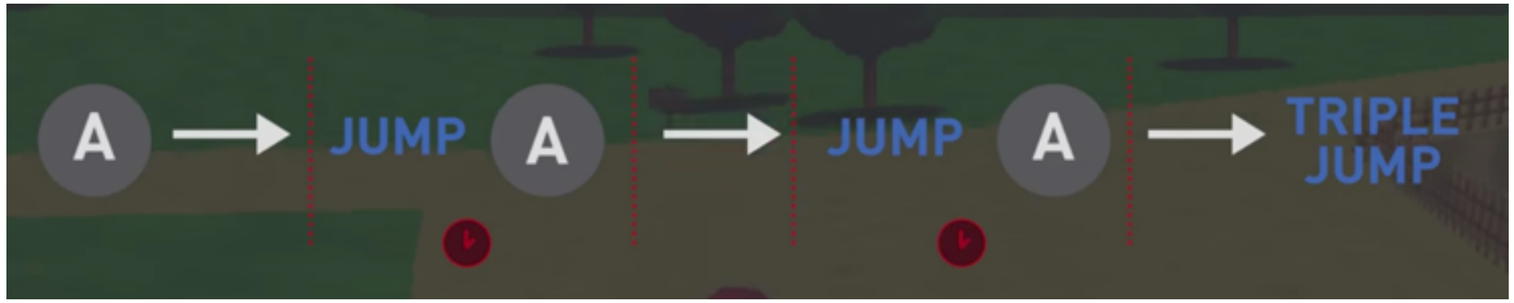
Modify another Verb

XOR Verbs (Push/Release)

Combine with Analog Device (Direction, Speed, Rotation)

Mario's Jump

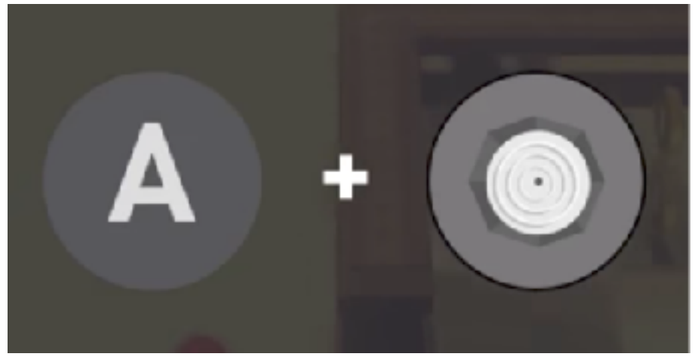
Duration



Multi Press

Modify

XOR



Analog

Combo

CHARACTERS

*Commands written here are for characters facing right. All commands are reversed when facing left.

NORMAL THROWS	
SHOULDER THROW	(CLOSE TO OPPONENT) ← or N + LP LK
SOMERSAULT THROW	(CLOSE TO OPPONENT) ← + LP LK
V-SKILL	
MIND'S EYE	MP MK
V-TRIGGER	
DENJIN RENKI	HP HK
V-REVERSAL	
HASHOGEKI	(DURING GUARD) → + 3P
UNIQUE ATTACKS	
COLLARBONE BREAKER	→ + MP
SOLAR PLEXUS STRIKE	→ + HP
AXE KICK	← + HK
JODAN NIRENGEKI	HP ▶ HK
JODAN SANRENGEKI	MP ▶ HP ▶ HK
SPECIAL MOVES	
HADOKEN ■	↓↘→ + P *CHANGE BY HOLDING BUTTON DURING V-TRIGGER
SHORYUKEN ■	→↓↘ + P
TATSUMAKI SENPUKYAKU ■	↓↙← + K
AIRBORNE TATSUMAKI SENPUKYAKU ■	(DURING VERTICAL OR FORWARD JUMP) ↓↙← + K
CRITICAL ART	
SHINKU HADOKEN	↓↘→↓↘→ + P
DENJIN HADOKEN	(DURRING V-TRIGGER) ↓↘→↓↘→ + P

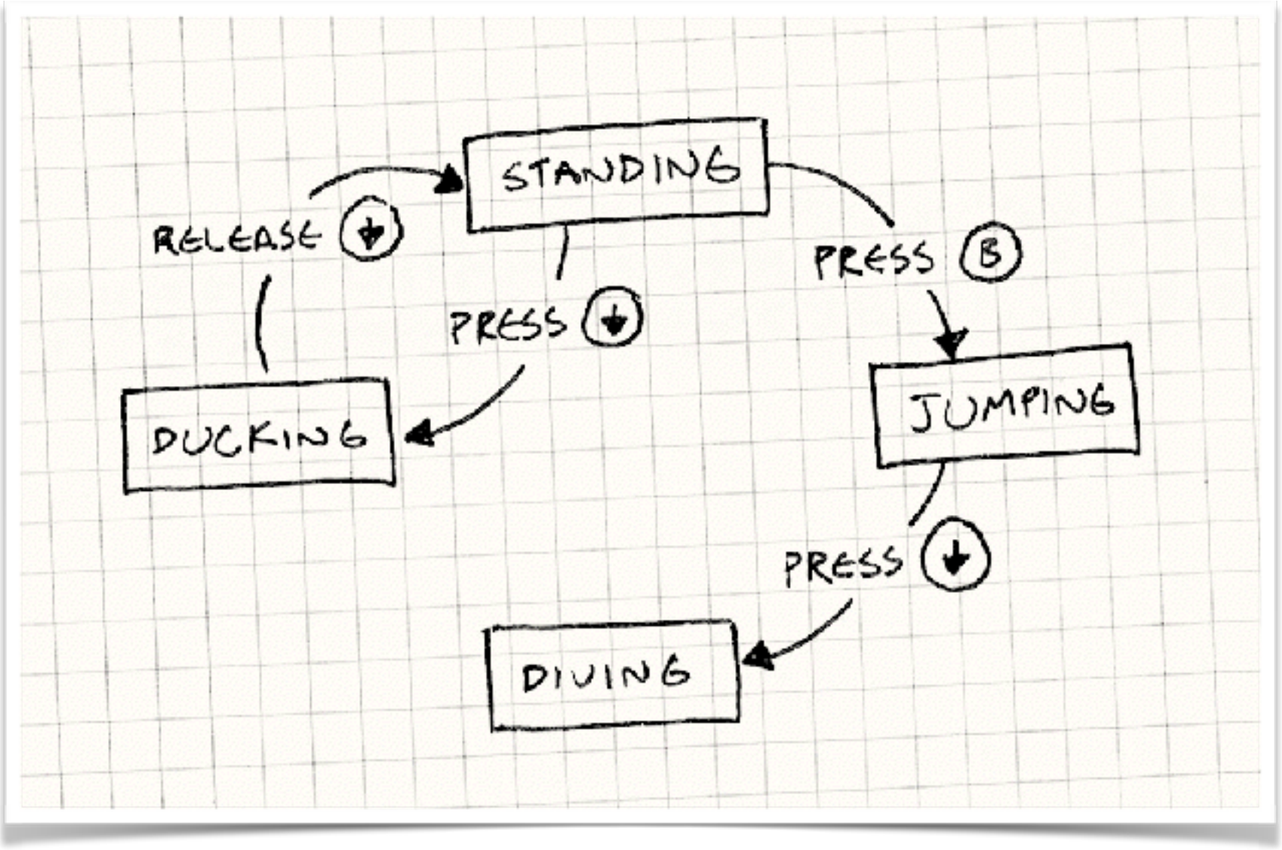
NORMAL THROWS	
KOSHUTO	(CLOSE TO OPPONENT) → or N + LP LK
TENSHIN SHUSHU	(CLOSE TO OPPONENT) ← + LP LK
RYUSEIRAKU	IN AIR LP LK
V-SKILL	
RANKYAKU	MP MK
V-TRIGGER	
RENKIKO	HP HK
V-REVERSAL	
SOHAKKEI	(DURING GUARD) → + 3P
UNIQUE ATTACKS	
TSUITOTSUKEN	← or → + MP
HAKKEI	← + HP
SENENSHU	↙ + MK
TENKUKYAKU	← + HK
YOKUSENKYAKU	→ + HK
KAKURAKUKYAKU	↙ + HK
YOSOKYAKU	(DURING JUMP) ↓ or ↙ + MK *UP TO 3 TIMES
WALL JUMP	(DURING JUMP NEAR WALL) ↗
SPECIAL MOVES	
KIKOKEN ■	←→ + P
HYAKURETSUKYAKU ■	↓↘ + K
AIRBORNE HYAKURETSUKYAKU ■	(DURING VERTICAL OR FORWARD JUMP) ↓↘ + K
SPINNING BIRD KICK ■	↓↗ + K
CRITICAL ART	
HOYOKUSEN	↓↘→↓↘→ + K

NORMAL THROWS	
DRAGON SUPLEX	(CLOSE TO OPPONENT) → or N + LP LK
TARGET DOWN	(CLOSE TO OPPONENT) ← + LP LK
AIR JACK	IN AIR LP LK
V-SKILL	
BULLET CLEAR	MP MK
V-TRIGGER	
SONIC MOVE - HIDE	HP HK
SONIC MOVE - BLITZ AIR	← + HP HK
SONIC MOVE - STEEL AIR	→ + HP HK
V-REVERSAL	
SONIC MOVE - AVOID	(DURING GUARD) → + 3P
UNIQUE ATTACKS	
KNEE BAZOOKA	→ + LK
JUMPING SOBAT	→ + MK
CHOPPING ASSAULT	→ + MP
STEP KICK	→ + HK
SPINNING BACK KNUCKLE	→ + HP
SIDE KNEE ATTACK	← + MK
RAPID PUNCH	LP ▶ MP
RAPID KICK	LK ▶ MK
WIND SHEAR	MP ▶ LK ▶ HP
DOWN BURST	↓ + MP ▶ → + MP
RAPTOR COMBINATION	MK ▶ HK ▶ MK
SPECIAL MOVES	
SONIC BOOM ■	↓↘→ + P *PRESS P DURING EX VER. FOR A SECOND ONE
SONIC SCYTHE ■	↓↙← + K
MOONSAULT SLASH ■	↓↘ + K
TRAGEDY ASSAULT ■	↙← + P
CRITICAL ART	
JUDGEMENT SABER	↓↘→↓↘→ + P

← : CHARGE ■ : EX VERSION EXISTS (COSTS 1 STOCK OF EX GAUGE) N : NO DIRECTION INPUTS

State Pattern

http://gameprogrammingpatterns.com/state.html



02. Combo Lab

Remember your fighting moves & customize your combos

Current PMP 101200

Pressen Unlock x 7

Active Combo

Combo 1

Y Y Y Y X Y Y X

Combo 2

Y X Y Y X X X X

Combo 3

X Y X Y Y Y X X

Combo 4

X X X Y Y X Y X

A Select B Back

Total Damage

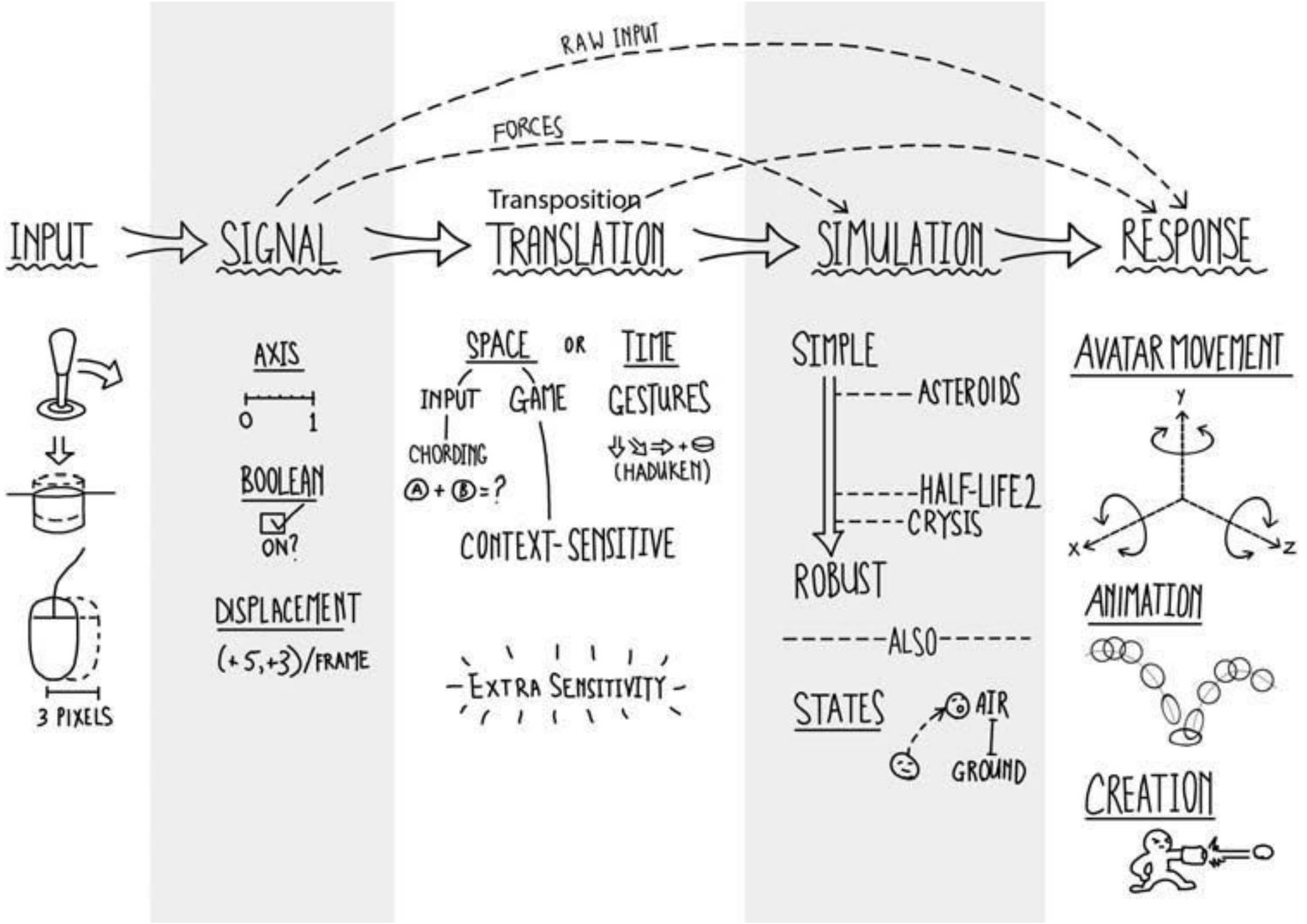
Regeneration

Cooldown Reduction

Pressen

SENSEN™

REMEMBER ME



Swink: Game Feel, S. 135