Game Technology

Lecture 1 – 17.10.2017
Input and Output

Dipl. Inform. Robert Konrad
Polona Caserman, M.Sc.

Prof. Dr.-Ing. Ralf Steinmetz
KOM - Multimedia Communications Lab

Hi

Dr. Stefan Göbel

- The boss

Robert Konrad

- Lecturer 1

Polona Caserman

- Lecturer 2

Dr. Florian Mehm

- Ex-Boss

Organization

TECHNISCHE UNIVERSITÄT DARMSTADT

Lecture (V2)

- Lecturers: Robert Konrad, Polona Caserman

Exercise (Ü2)

- Theory and implementation (game engine programming)

Language

- Answers are accepted in German and English (exercises and exam)

Organization

Sign up with TuCan

Current news

- Website@KOM (static information only): https://www.kom.tu-darmstadt.de/teaching/current-courses/gametech-lecture/overview1/
- Wiki, including the lecture slides and script:

  wiki.ktxsoftware.com

- Fachschafts-Forum:

  https://www2.fachschaft.informatik.tu-darmstadt.de/forum/viewforum.php?f=557

- **game-technology@kom.tu-darmstadt.de**

Exercises

Released after each lecture

- First exercise will be a special case, intended to bring everyone up to speed with git repositories, engine, …

Exercises will have due dates

- These dates are non-negotiable

Bonus Points

- >50%: 0.3; >70%: 0.7; >90%: 1.0
- The exam has to be passed without the bonus points – bonus is added only after the exam has been passed regularly
  - The bonus is applied by linearly interpolating
- Your bonus points will be uploaded to your git repository

Exercises

Group Exercises

- Allowed to complete exercises in groups up to **3 members**
- Turn in exercises via git until the noted time

Group Formation (1-3 people – please form teams!)

- Choose your own name
- Send group information to game-technology@kom.tu-darmstadt.de, including:
  - Group name
  - Names of all members
  - Mail adresses of all members
- Until Friday, October 20th, 23:59

Turning in Solutions

- Theory: Digital, scan written answers or work digitally (PDF, txt, …)
- Source Code: See C++ lecture part

Exercises

1 exercise per week

- Due until the next lecture
- No exercises during winter break

Git

- Instructions are sent with your group login

Relation to other lecturers

Serious Games

- Lecture
- Seminar
- (Projekt)Praktikum

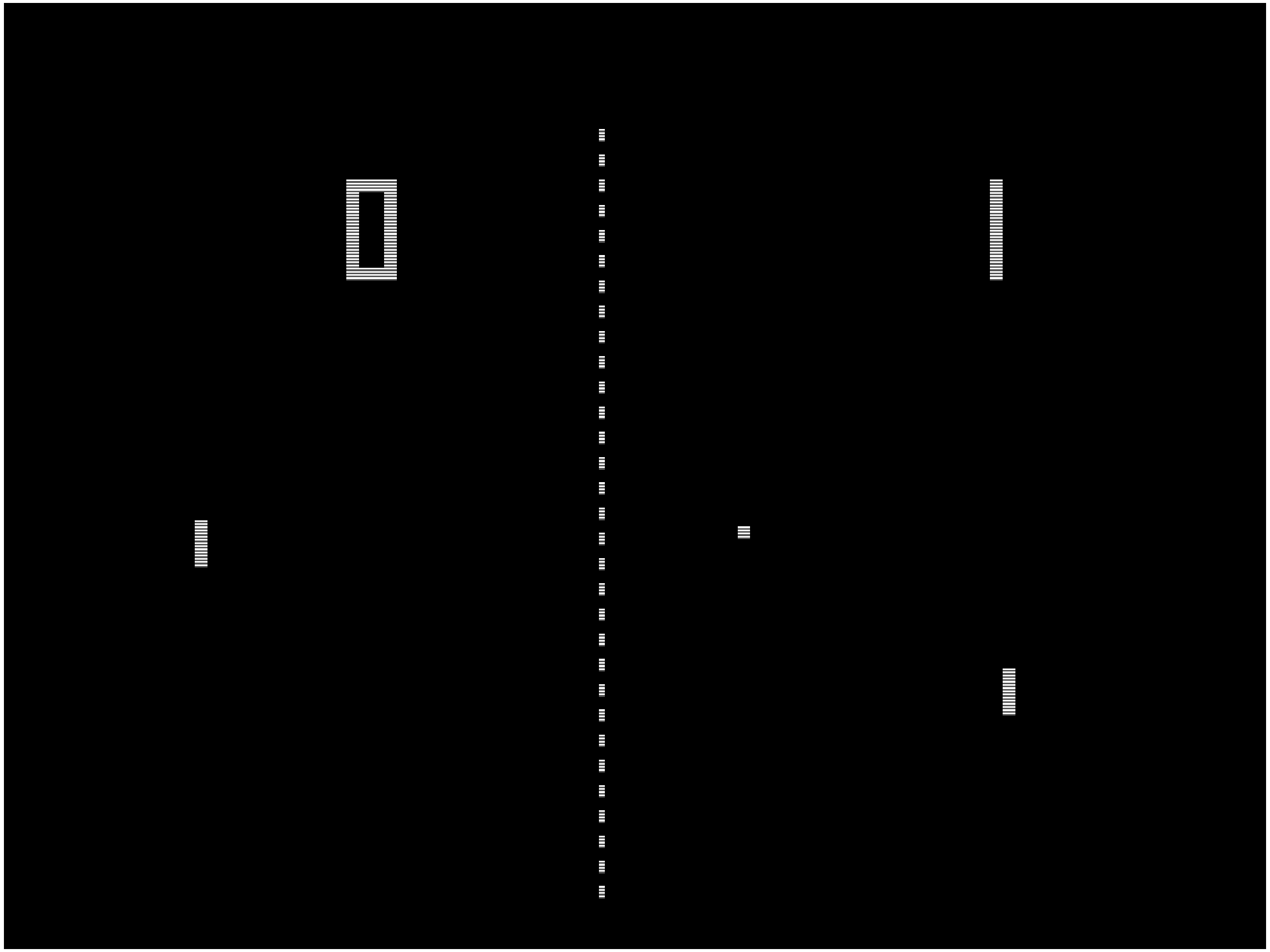
Urban Health Games

FIF Schwerpunkt Serious Games

- http://www.fif.tu-darmstadt.de/fif_topics_structure/fif_serious_games_structure_ref/index.de.jsp

Computer Graphics

Video Games



Pong, 1972

Focus on Performance

Manual memory management

- Pre-loading
- Cache optimization

Shader Programming

Separate lecture part for some lectures

~1 hour theory

~30 minutes programming, technology (e.g. GPU)

Motivation

Shaded Pixels per Second

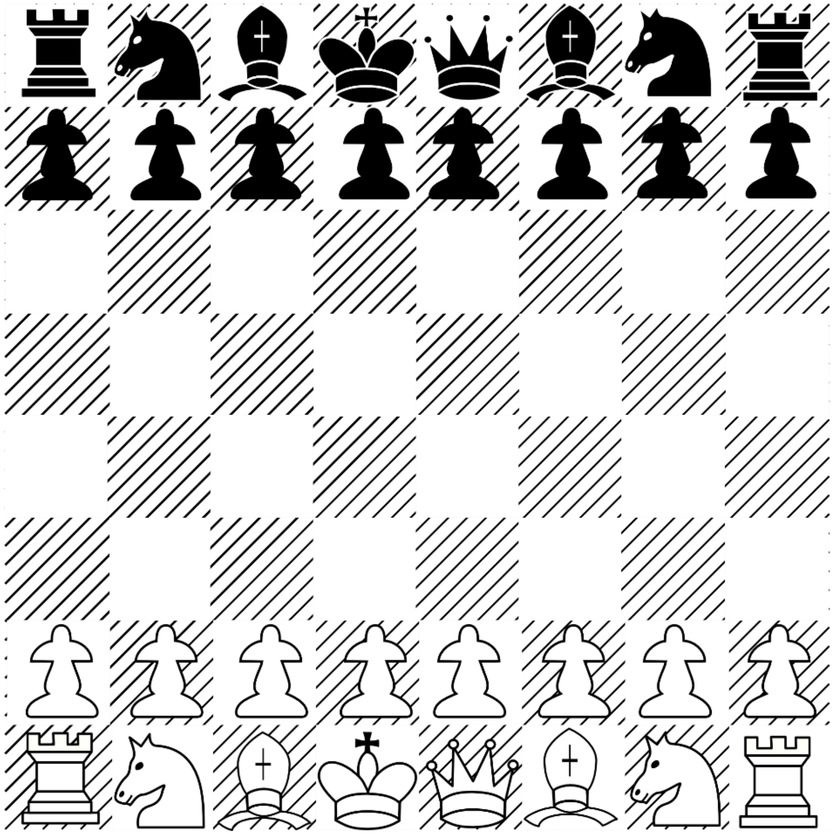
- 720p @ 30 Hz: 27 million pixels/sec
- 1080p @ 60 Hz: 124 million pixels/sec
- 30” Monitor 2560x1600 @ 60 Hz: 245 million pixels/sec
- VR 1512x1680x2 @ 90 Hz: 457 million pixels/sec
- 4k Monitor 4096x2160 @ 60 Hz: 530 million pixels/sec

Pseudo-realistic realtime simulations

Pseudo-realistic realtime simulations

No chess

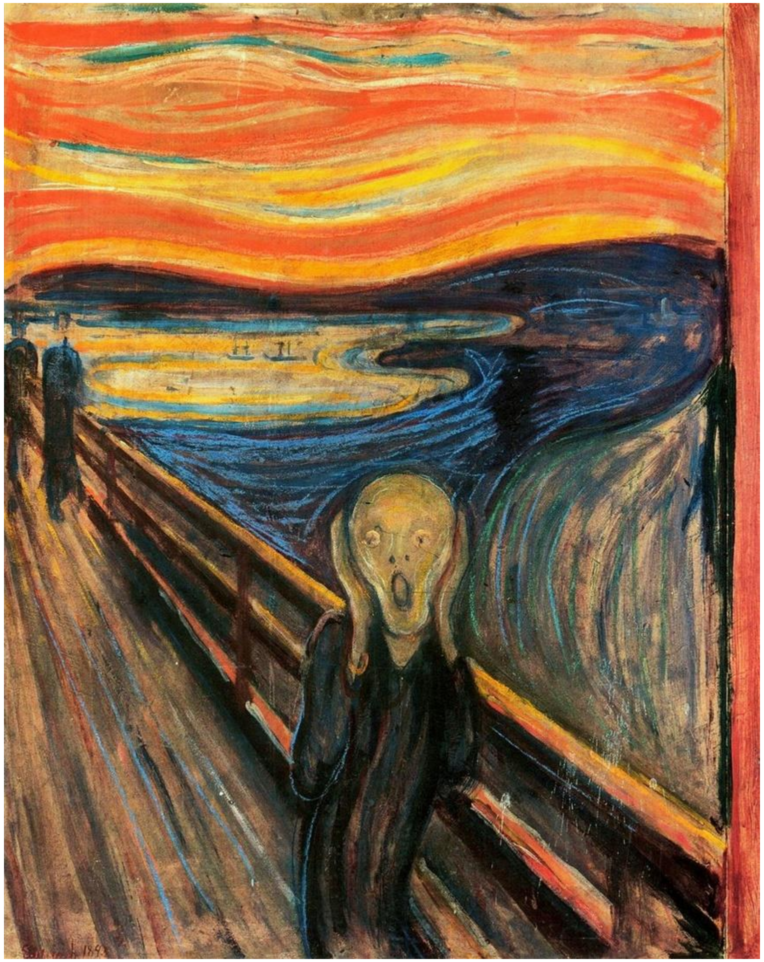
- Focus on fast/realtime apps
- Running in a game loop



Pseudo-realistic realtime simulations

No „artsy“ games

- But understanding how to make realistic games also helps with non-realistic games



Pseudo-realistic realtime simulations

No flight simulators for Lufthansa

- Actual realism not necessary
  - …and probably too slow
- Requires knowledge of human perception

Human-Machine data transfer

Human

- Output
  - Pushing
  - Talking
  - Moving
- Input
  - Staggering amounts of data

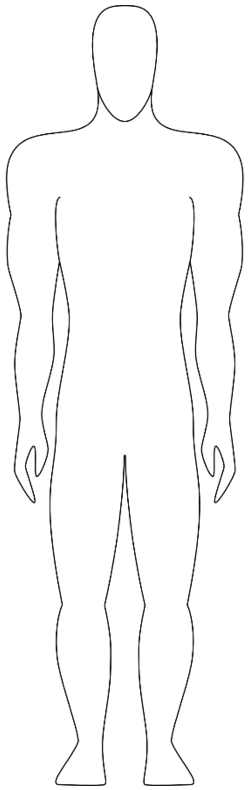
Machine

- Output
  - Monitor
  - Speakers
- Input
  - Buttons

Humans

Five senses

- Sight
- Hearing
- Touch
- Smell
- Taste

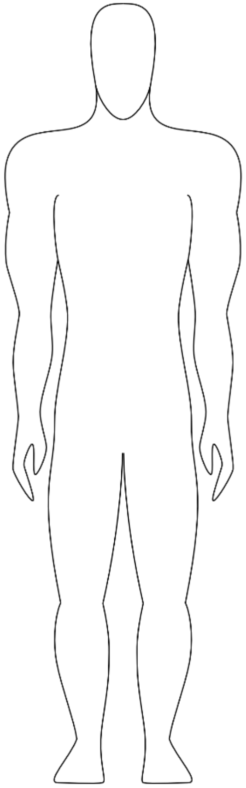


Humans

TECHNISCHE UNIVERSITÄT DARMSTADT

Many senses

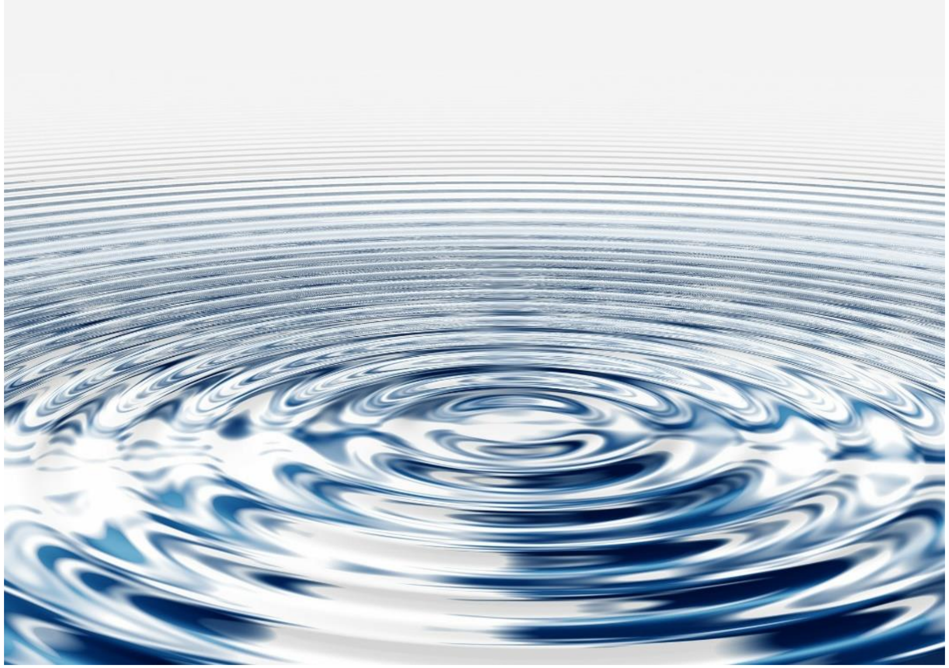
- External
  - Sight
  - Hearing
  - Touch
  - Smell
  - Taste
  - Acceleration
  - Temperature
- Internal
  - Kinesthetic
  - Pain
  - …



Eyes and Ears

Most dominant sensors

Measure different kinds of waves



Waves

Wave Direction

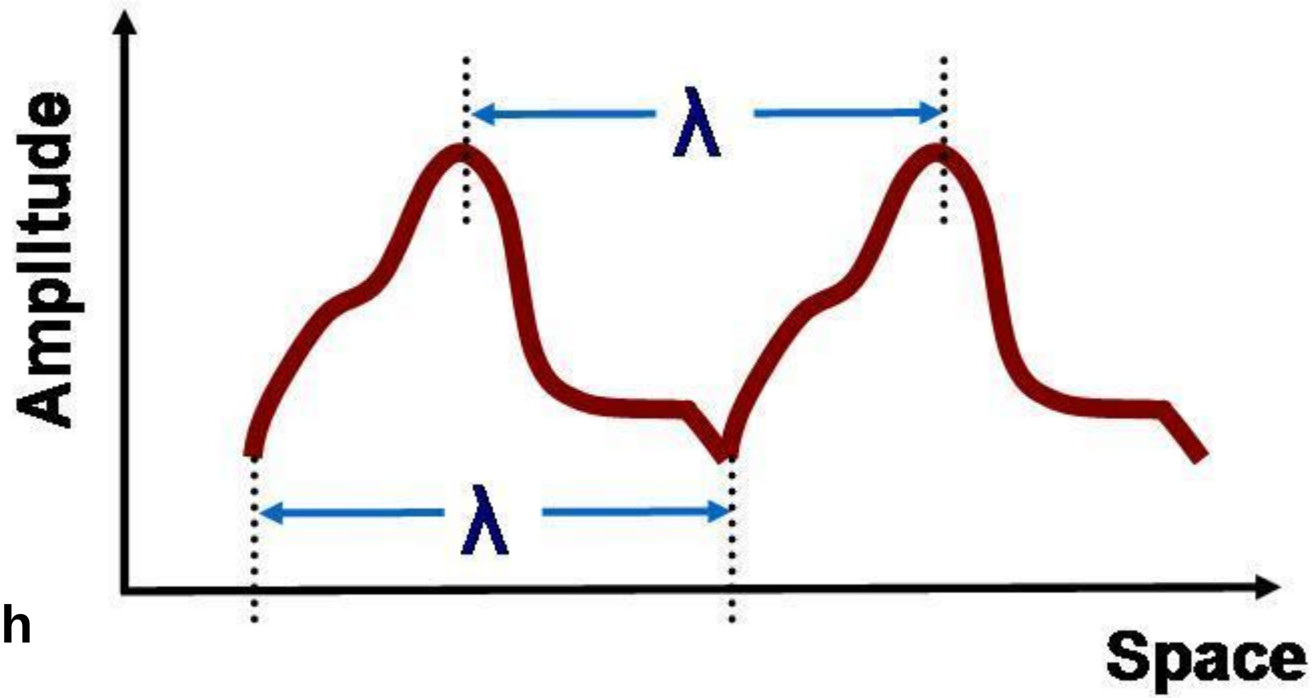
Oscillation Direction (for transverse waves)

Amplitude

Speed (often constant)

Wavelength

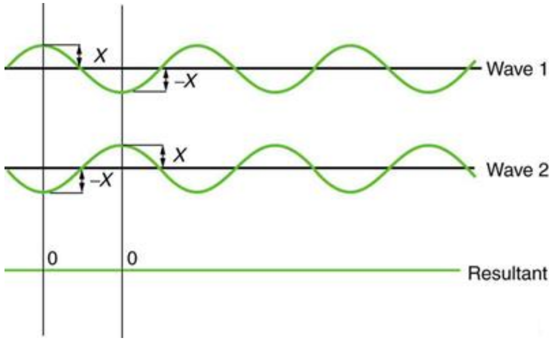
Waveform

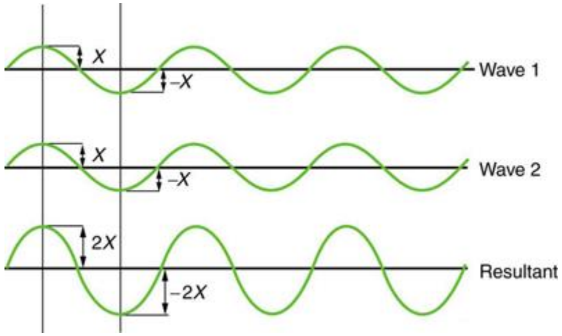


Frequency =
Speed / Wavelength

Wave Interaction

Superposition





Light Waves

TECHNISCHE UNIVERSITÄT DARMSTADT

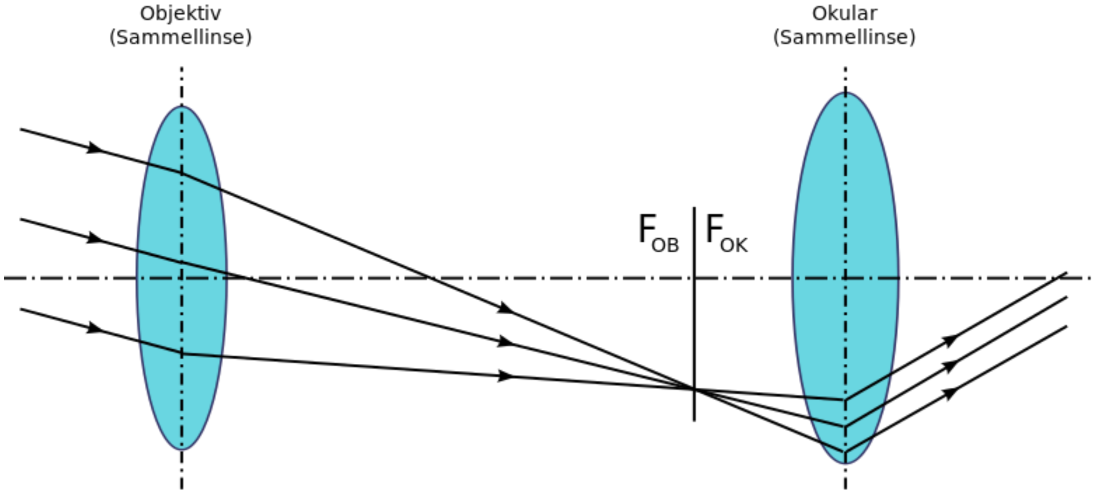
Electromagnetic waves

Transverse waves

- Direction of oscillation orthogonal to wave direction

Very fast

Usually discussed using simplified models



Optical Sensors

Two units

- Surround view or 3D view depending on arrangement

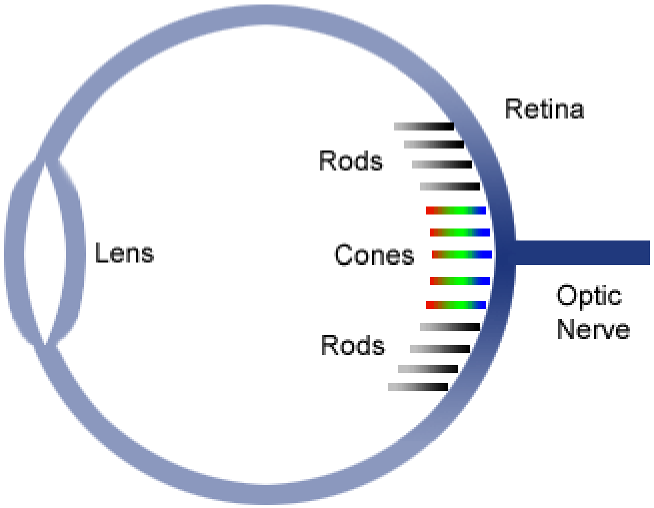
The eye

The lens focuses light on the retina

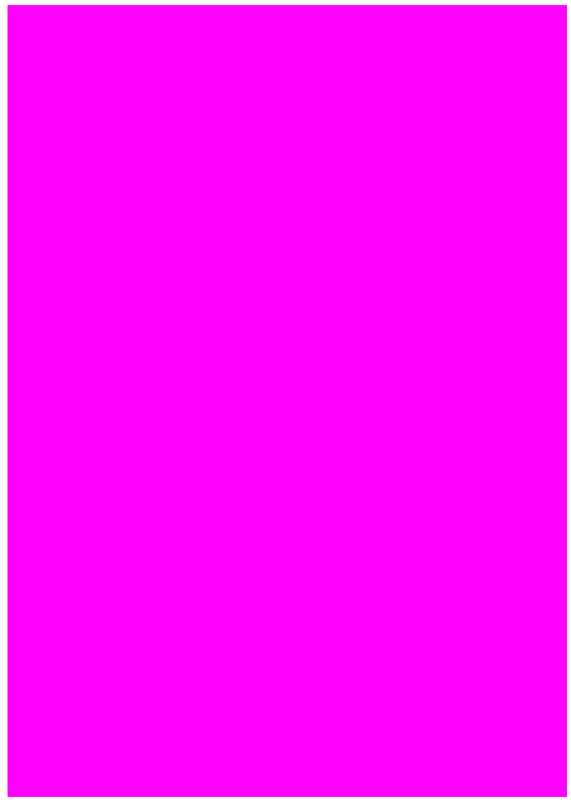
Rods measure light intensity/energy (wave amplitude and frequency)

Cones only react to specific wavelengths

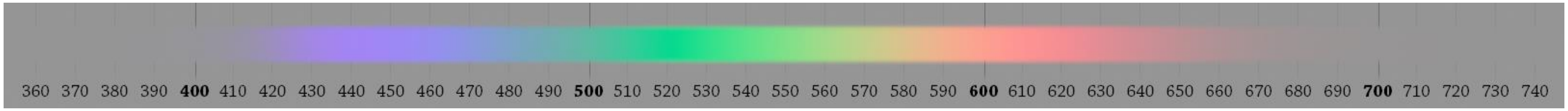
- Three different kinds
  - Red,
  - green, and
  - blue



What do you see?



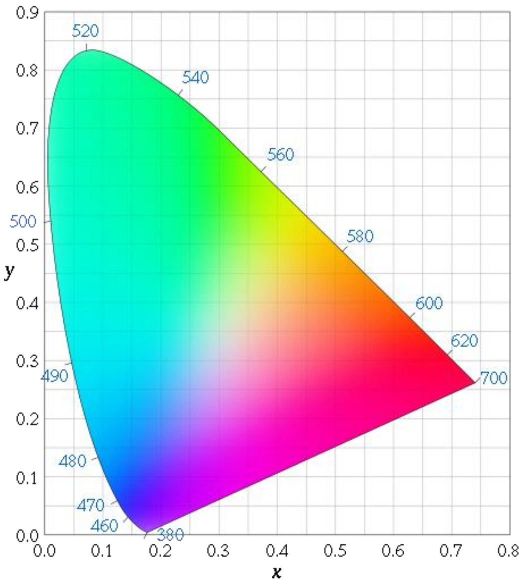
Red, green and blue



Brain interpolates colors

Brain sees magenta when interpolation fails

- Same amounts of blue and red but no green
- See http://richannel.org/colour-mixing-and-the-mystery-of-magenta



Visual Field of Humans

Horizontally: ~180°

Vertically: ~135°

But, the vision quality is not the same across the visual field

- Binocular vision: ~135°
  - Remaining visual field only visible by one eye
- Color vision
  - Cones mostly in the center of the field of view → good color vision
  - Rods mostly on the periphery → good shape perception

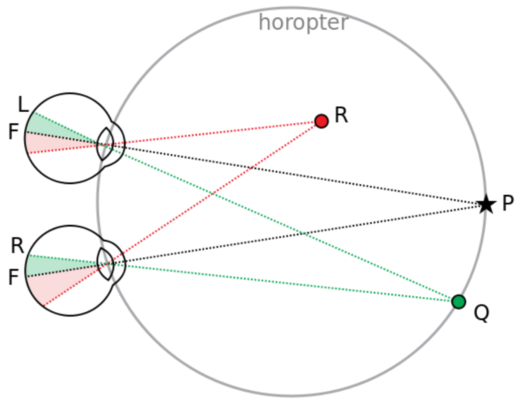
Foveated rendering

- Track what the eyes are focusing
- Reduce detail in the periphery →speed up

Stereo Vision, Depth Perception

Distance between eyes

- Interpupillary Distance
- ~6.5 cm in humans



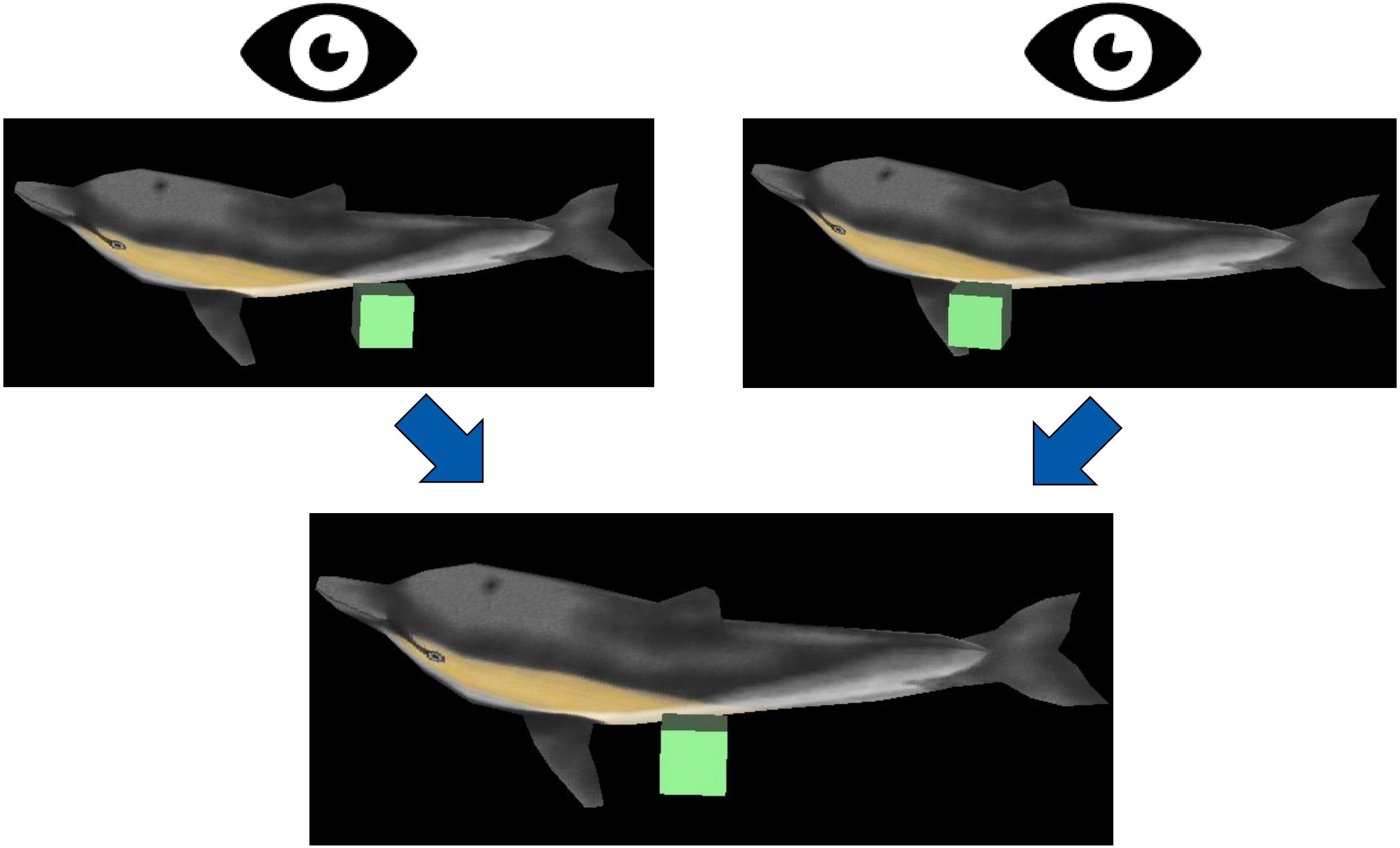
Monocular cues

Binocular cues

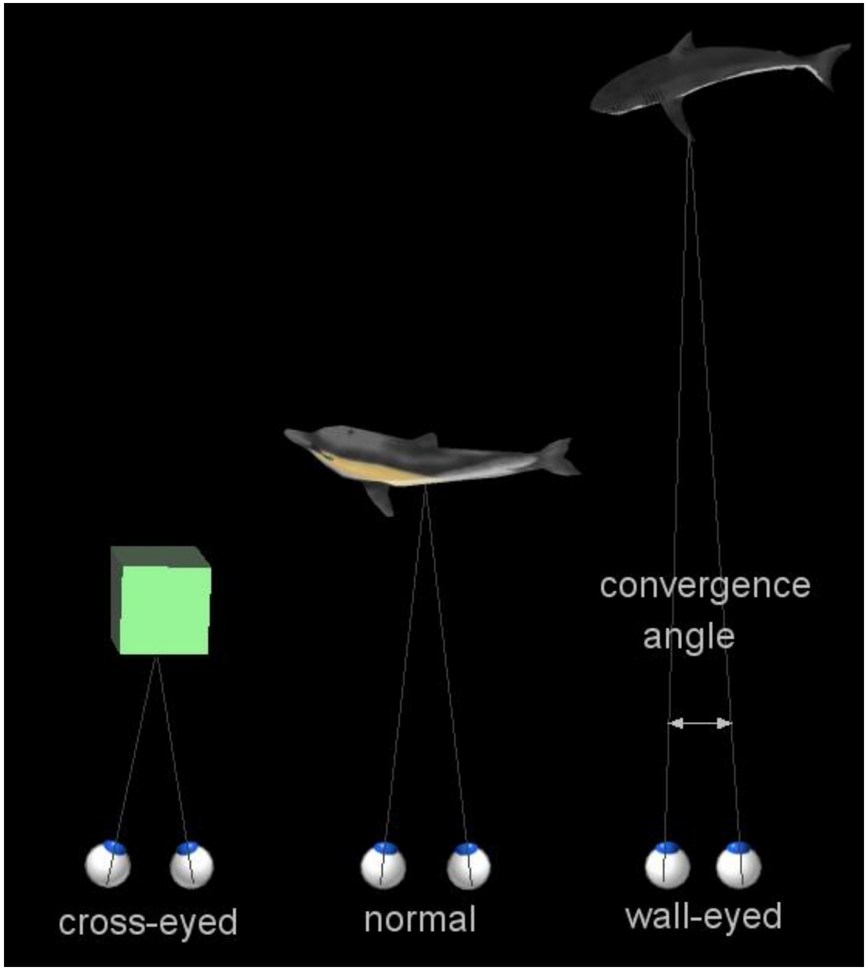
- Stereopsis: Triangulation using difference in both eyes (effective for < 200 m, differs according to sources)
- Convergence: Using muscles in the eyes (effective for < 10 m)
- Shadow Stereopsis

→ Limits to distances, opens doors for optimization in VR

Stereopsis



Convergence

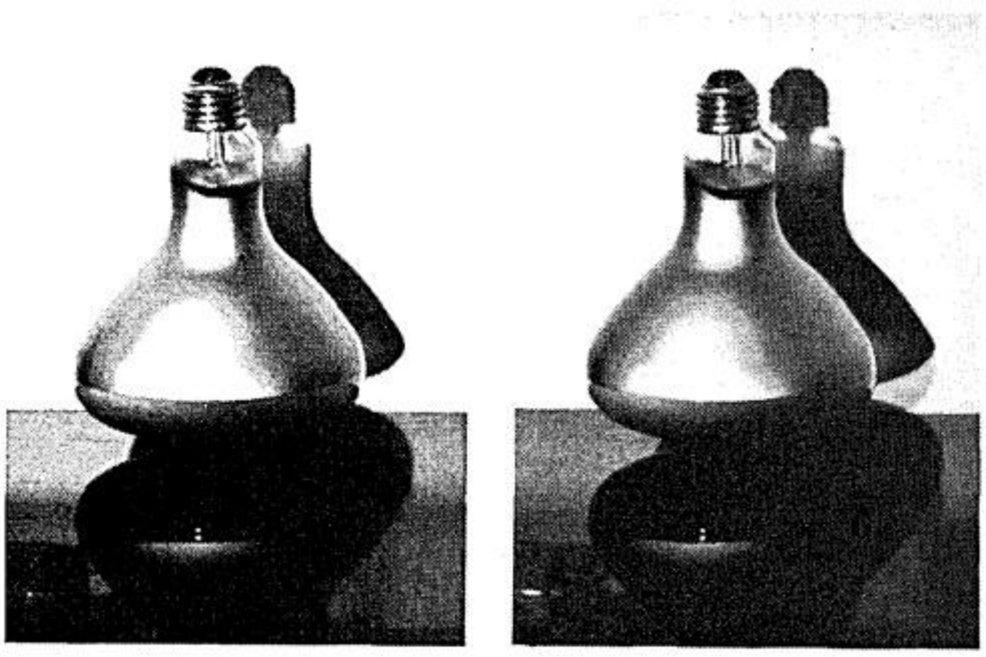


https://en.wikipedia.org/wiki/Vergence#Convergence

Shadow Stereopsis

Antonio Medina Puerta, "The power of shadows: shadow stereopsis," J. Opt. Soc. Am. A 6, 309-311 (1989)

Images with no parallax disparities but shadow differences still appear to have depth



Stereo Vision, Depth Perception

Monocular Cues

- Motion parallax
- Depth from motion
- Kinetic depth effect
- Perspective
- Relative size
- Familiar size
- Absolute size
- Accommodation
- Occlusion
- Curvilinear perspective
- Texture gradient
- Lighting and shading
- Defocus blur
- Elevation

Stereo Vision, Depth Perception

Monocular Cues

- **Motion parallax**
- Depth from motion
- Kinetic depth effect
- Perspective
- Relative size
- Familiar size
- Absolute size
- Accommodation
- Occlusion
- Curvilinear perspective
- Texture gradient
- Lighting and shading
- Defocus blur
- Elevation

Motion Parallax

Objects at different distances appear to move at different speeds when moving relative to the observer

Ninja Gaiden II, 1990

https://www.youtube.com/watch?v=Mem4xQAhDfE

What is missing?

„A little house“ by Dono: http://www.blendswap.com/blends/view/4500

Stereo Vision, Depth Perception

Monocular Cues

- **Motion parallax**
- Depth from motion
- Kinetic depth effect
- Perspective
- Relative size
- Familiar size
- Absolute size
- Accommodation
- Occlusion
- Curvilinear perspective
- Texture gradient
- **Lighting and shading**
- Defocus blur
- Elevation
- Aerial Perspective

Lighting and Shading

Stereo Vision, Depth Perception

Monocular Cues

- **Motion parallax**
- Depth from motion
- Kinetic depth effect
- Perspective
- Relative size
- Familiar size
- Absolute size
- Accommodation
- Occlusion
- Curvilinear perspective
- Texture gradient
- **Lighting and shading**
- Defocus blur
- Elevation
- **Aerial Perspective**

Aerial Perspective

Due to the influence of the atmosphere, objects far away appear subdued and look more and more like the horizon

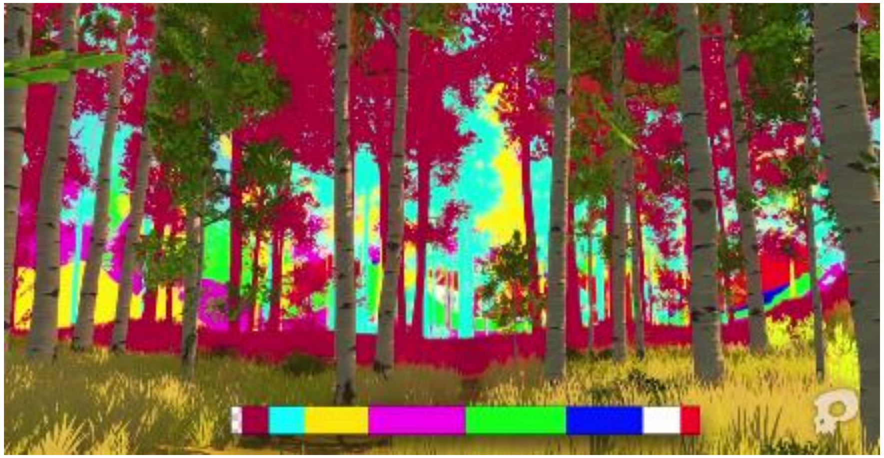
Der Wanderer über dem Nebelmeer, Caspar David Friedrich, 1818

Aerial Perspective

Used formerly as performance optimization

Silent Hill, 1999

Nowadays, more artistic choice



Firewatch, 2016

Stereo Vision, Depth Perception

Monocular Cues

- **Motion parallax**
- Depth from motion
- Kinetic depth effect
- Perspective
- Relative size
- Familiar size
- Absolute size
- Accommodation
- Occlusion
- Curvilinear perspective
- **Texture gradient**
- **Lighting and shading**
- Defocus blur
- Elevation
- **Aerial Perspective**

Texture Gradient

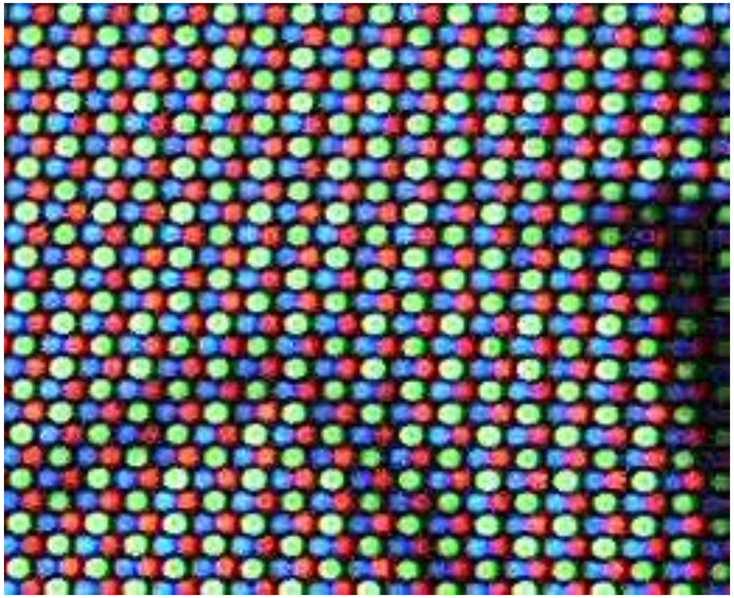
Regular patterns get more densely packed the further they are away

Monitors

Exact counterpart to human eye

Red, green and blue emitters

No physically accurate picture reproduction



Computer → Monitor

Designated memory area which is transferred to the monitor

- The framebuffer

Structurally equivalent to the pixel structure

- 1 red byte
- 1 green byte
- 1 blue byte, …

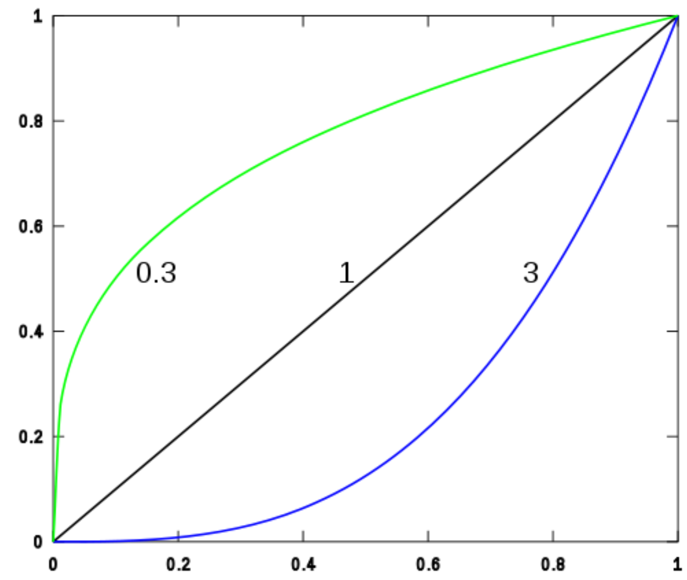
Gamma

TECHNISCHE UNIVERSITÄT DARMSTADT

Monitors do not emit 50% light intensity for a 50% light value (neither do our eyes work linearly)

Work according to a gamma function

$$I_{out} = {I_{in}}^{\gamma}$$



Monitor color space is not ideal for lighting calculations

Usually we choose $\gamma = 2.2$

More info: http://http.developer.nvidia.com/GPUGems3/gpugems3_ch24.html

Gamma

If images are saved non-linearly, we can encode tones better to match human vision

- Human eyes are more sensitive for differences in darker tones

Original: Values from 0 to 1

Linearly encoded (using 4 bits)

Gamma corrected (using 4 bits)

Gamma correction

Input from gamma-corrected images

- Raise values to the power of $\boldsymbol{\gamma}$
- Note: Can be done with integer (e.g. 0 – 255) or floating point values (0.0 – 1.0)

→Brings colors into linear space

Handle calculations in linear space

Output to the monitor

- Raise output values to the power of $\frac{1}{\gamma}$
- If needed, clamp to minimal and maximal value (e.g. 0 and 255)

→Brings colors into gamma-corrected space

Sound Waves

Air compression

Longitudinal Waves

~343 m/s

Sound Sensors

Also two units

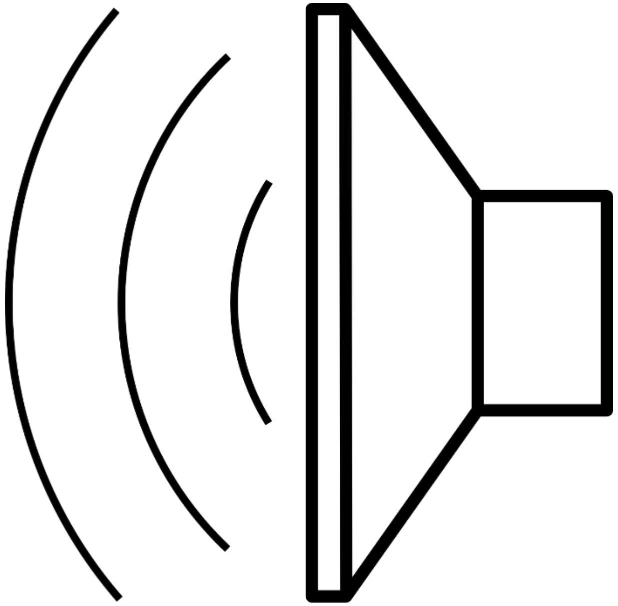
Infer direction by measuring time differences

Measure actual wave forms

Loudspeakers

Construct actual sound waves

Physically accurate reproduction of original waves



Computer → Speaker

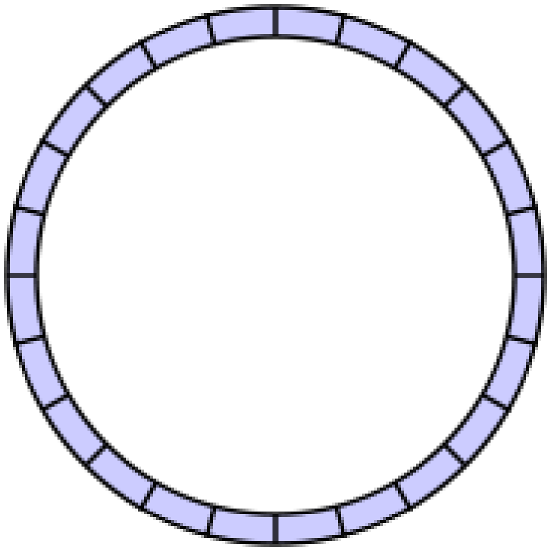
Small ring buffer

- Write samples into the buffer
- Read back during playback

Discretely sampled waveform

Pointer to last sample written

Pointer to next sample to read



Sound Mixing

Superpositioning

- Adding waves

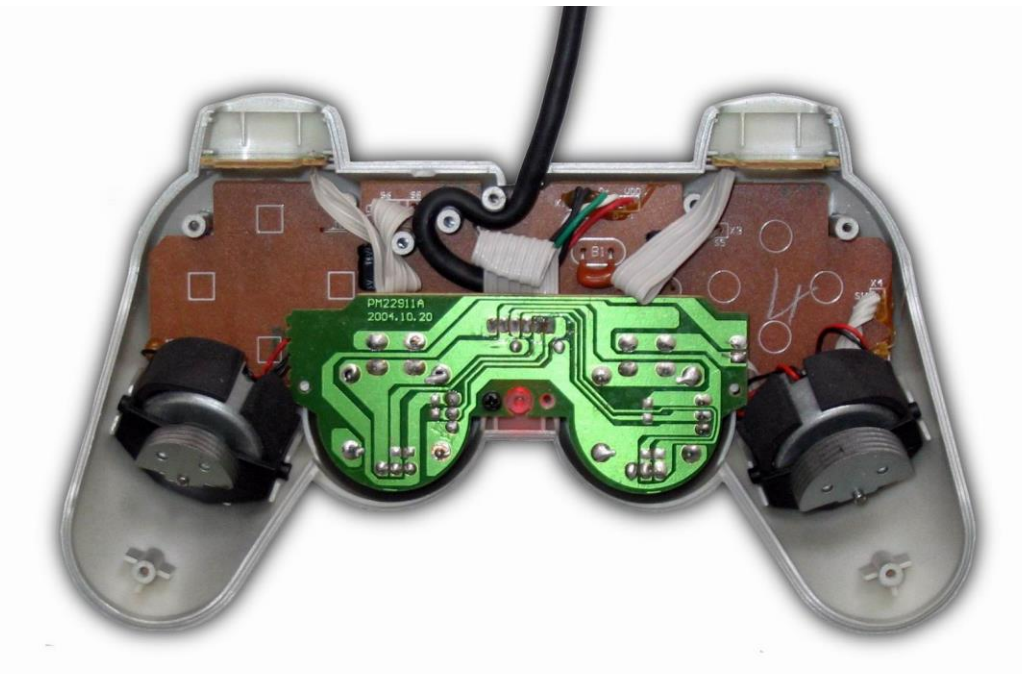
Again physically accurate

Actual danger of superposition effects

- Avoid mixing identical sounds
- In reality, events rarely/never happen at the exact same time

Rumble / Force Feedback

Very restricted „touch“ output



Acceleration output

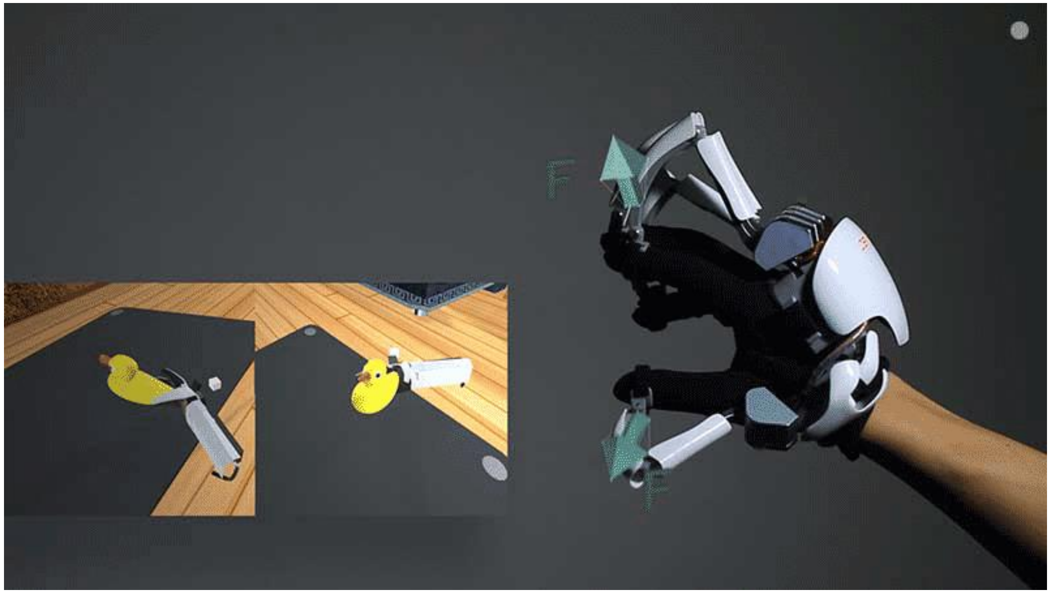
Sega R-360, 1991

Kinesthetic

Virtuix Omni, 2015

Exoskeletons

- Dexmo Glove, 2016



Computer input

TECHNISCHE UNIVERSITÄT DARMSTADT

Mouse, Keyboard, Gamepad, …

Mostly trivial

Important to reduce input lag

- Minimize time from input to output
-

Nintendo Power Glove, 1989

Complex computer input

Input inaccuracies

- Compensate by being overly optimistic

Practical Part

C

Portable assembler

Developed for/with UNIX

From 1969

Dennis MacAlistair Ritchie (September 9, 1941 – October 12, 2011)

C/C++

Open standards, not bound to a company

Available almost anywhere

- Even in the browser (Emscripten/WebAssembly)

Bjarne Stroustrup (*30.12.1950)

C++

Adds higher level concepts to C

No performance regressions

Originally „C with classes“

From 1979

Classes

```
class Foo {
public:
  Foo() {
        x = 2;
  }
private:
  int x;
};
```

Free functions

```
int main(int argc, char** argv) {
  return 0;
}
```

Main entry point

- But not on every system

*** is a pointer**

- A memory address

char* is used for strings

char - multiple strings**

Header files

Using multiple source files is complicated

Compiler compiles single cpp file to object file

- Files can #include other files in a preprocess
- Use separate, minimal header files for #include

A separate linker application links multiple object files

No standard to tell the linker what to do

Primary reason that compiling C/C++ is slow

Foo.h

```
#pragma once

class Foo {
public:
  Foo();
private:
  int x;
};
```

#pragma once is not part of the standard, but widely adopted

- Easier to write and read than other way of include guards

Foo.cpp

```
#include "Foo.h"

Foo::Foo() {
  x = 2;
}
```

C++ in 20XX

Very big language

Complex features

- Templates (similar to Java‘s generics) are turing complete

Contains fancy library

- Automates memory management somewhat
- std::string, std::vector, …

boost Library

- Widely used
- Big, std style library

C++ in 20XX

TECHNISCHE UNIVERSITÄT DARMSTADT

Very big language

Complex features

- Templates (similar to Java‘s generics) are turing complete

Contains fancy library

- Automates memory management somewhat
- std::string, std::vector, …

boost Library

- Widely used
- Big, std style library

> **John Carmack**
> @ID_AA_Carmack
>
> Saw comment // NEW BOOST CODE, and had a moment of panic before realizing it was vehicle boost, not C++ boost

Hardware Access

Files

- That‘s it

No support for

- Special directories
- Memory mapped files
- …

OpenGL

Standard API for Graphics Hardware

Many different versions

Not on consoles

- In general similar to desktop variants, but specific to the capabilities of the one GPU in question

Questionable support by Apple and Microsoft

http://www.eurogamer.net/articles/digitalfoundry-how-the-crew-was-ported-to-playstation-4

GPU Programming Languages

GLSL

- Part of OpenGL

HLSL

- Microsoft (Direct3D and Xbox)
- Sony (all PlayStations)

Metal

Apple

Audio, Keyboard

Practically no standards

SDL can do the job

- Simple DirectMedia Layer
- https://www.libsdl.org/

Kore

- **APIs for**
  - Graphics (encapsulates OpenGL and DirectX)
  - Audio
  - Input Devices
  - File Access
  - …

- **GLSL cross compiler**

- **https://github.com/Kode/Kore**

- **Introductions at http://wiki.ktxsoftware.com**