Game Technology

Lecture 12 – 23.01.2018
Audio and Scripting

TECHNISCHE UNIVERSITÄT DARMSTADT

Dipl-Inf. Robert Konrad
Polona Caserman, M.Sc.

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

Sound Waves

TECHNISCHE UNIVERSITÄT DARMSTADT

Sound waves

- Air compression
- Longitudinal Waves
- ~343 m/s
- 20 to 17000 Hz

Loudspeakers

TECHNISCHE UNIVERSITÄT DARMSTADT

Converts electrical signals to sound waves

- Using an acoustic membrane

Ears

Two identical audio sensors

Measures actual wave forms

- Using the eardrum

Computer -> Speaker

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Small ring buffer**
- **Discretely sampled waveform**
- **Pointer to last sample written**
- **Pointer to next sample to read**

Sound Mixing

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Superpositioning**
  - Adding waves
- **Physically accurate**
- **Actual danger of superposition effects**
  - Avoid mixing identical sounds

Music

Music

TECHNISCHE UNIVERSITÄT DARMSTADT

Long files

- Played/Streamed in the background
- Mostly not influenced by gameplay

Sound Effects

TECHNISCHE UNIVERSITÄT DARMSTADT

Short files

- Triggered by gameplay
- Modified according to position, environment,…

Speech

TECHNISCHE UNIVERSITÄT DARMSTADT

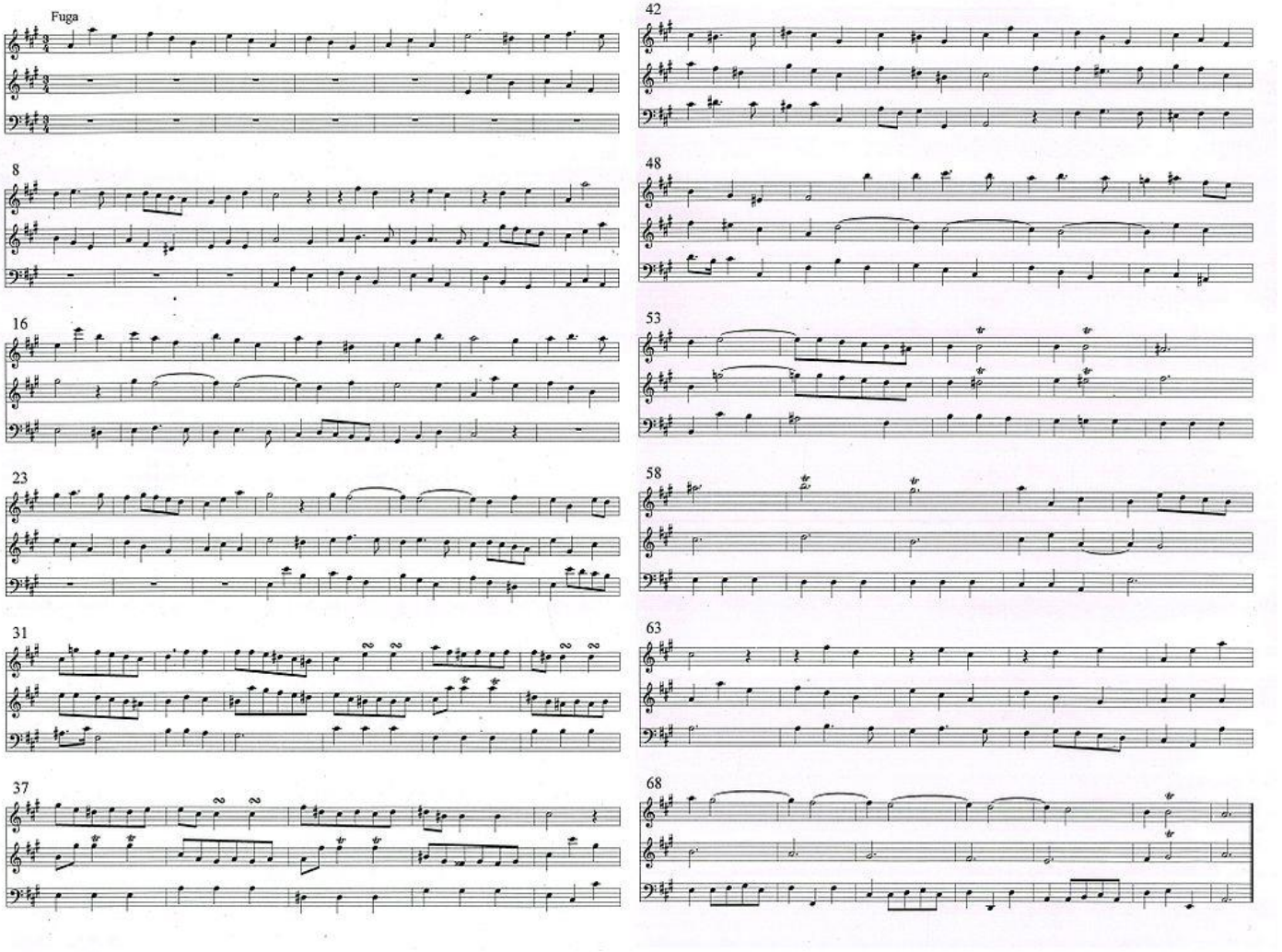
Sometimes more like sound effects

- „Ouch“

Sometimes more like music

- „lalalalalala“
- „blablablabla“

Music



Music

TECHNISCHE UNIVERSITÄT DARMSTADT

Pitch

- Frequency
- c d e f g a h c

Duration

- Duration

Loudness

- ~Amplitude

Tone Color

- Wave form
- Instrument

Early 80s Music

TECHNISCHE UNIVERSITÄT DARMSTADT

Early games used simple wave forms

- Square waves
- Triangle waves
- Sawtooth waves
- Plus noise

NES

Game Boy

Master System

…

http://studio.substack.net

```
return function (t) {
    function sin (x) { return Math.sin(2 * Math.PI * t * x) }
    function square (x) { return (Math.floor(x) % 2 == 0) ? -1 : 1; }
    function saw (x) { return (-x % 1 - 0.5) * 2; }
    function triangle (x) { return 2 * Math.abs(2 * (x - Math.floor(x + 0.5))) - 1; }

    //return sin(441);
    //return square(t * 200) / 20;
    //return saw(t * 240) / 10;
    return triangle(t * 440) / 3;
}
```

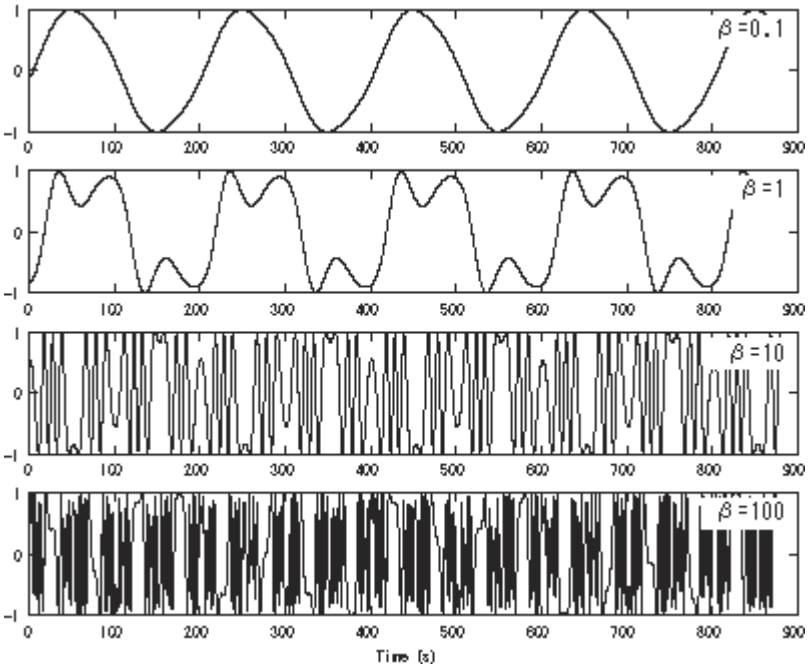
Late 80s / Early 90s Music

TECHNISCHE UNIVERSITÄT DARMSTADT

FM-Synthesis

Modulator (usually sin) drives the Carrier

$$s_{fm}(t) = A \cdot \cos\left(\omega_c t + \beta \cdot \cos(\omega_m t)\right)$$



AdLib

Mega Drive

Also 80s synthie music

http://greweb.me/2013/08/FM-audio-api/

Early 90s

TECHNISCHE UNIVERSITÄT DARMSTADT

„Tracker Music“

„Module Files“

Use short sound samples to represent instruments

- Change pitch as needed (or use more samples)

Amiga

SNES

(MT-32, General MIDI)

Gravis Ultrasound

Dynamic Music

iMuse

Banjo Music

CDs

CDs

TECHNISCHE UNIVERSITÄT DARMSTADT

Plays music

No application control

- Apart from start/stop

No file loading while music plays

WAV, MP3,…

TECHNISCHE UNIVERSITÄT DARMSTADT

Play back large sound files manually

More flexible than CD audio

- But not as flexible as previous methods

Today audio compression is widely used

Orchestras

Sequencer

Sequencer

TECHNISCHE UNIVERSITÄT DARMSTADT

Basically works like old tracker programs

- But more and bigger samples, more effects,…

But almost always exports only to wav

Sound effects back then

TECHNISCHE UNIVERSITÄT DARMSTADT

Originally based on square waves,…

http://www.bfxr.net

Sound effects now

TECHNISCHE UNIVERSITÄT DARMSTADT

Recorded samples

Little to no hardware support

Number of simultaneously mixed samples usually limited

Sound Localization

TECHNISCHE UNIVERSITÄT DARMSTADT

Distance

- Increased distance -> Decreased amplitude (amplitude *= 1 / distance)
  - (and slightly decreased frequency)

Direction

- Interpolate between speakers
- Better quality -> add more speakers

Dolby Atmos,…

TECHNISCHE UNIVERSITÄT DARMSTADT

No fixed number of speakers (as compared to Dolby Digital 5.1,…)

Define sounds sources with positions

Audio system maps to actual speakers (or headphones)

(Audio3 in Kore)

Sound Direction

TECHNISCHE UNIVERSITÄT DARMSTADT

Can also be simulated using headphones

Brain analyzes sounds to infer directions

https://www.youtube.com/watch?v=8IXm6SuUigI

Sound Localization Left/Right

TECHNISCHE UNIVERSITÄT DARMSTADT

Measure time differences between ears

Loudness differences between ears

- Because of the head
- Depends highly on frequency
  - Partly used for frequencies > 800 Hz
  - Exclusively used for frequencies > 1600 Hz

Sound Localization Front/Back

TECHNISCHE UNIVERSITÄT DARMSTADT

Analyzes frequency differences caused by the ear forms

- Also to a lesser degree head, shoulders,…
- Sadly somewhat individual

Analyzes changes due to head movements

Doppler Effect

TECHNISCHE UNIVERSITÄT DARMSTADT

Frequency increases/decreases when sound source/receiver moves

For decreasing/increasing distance

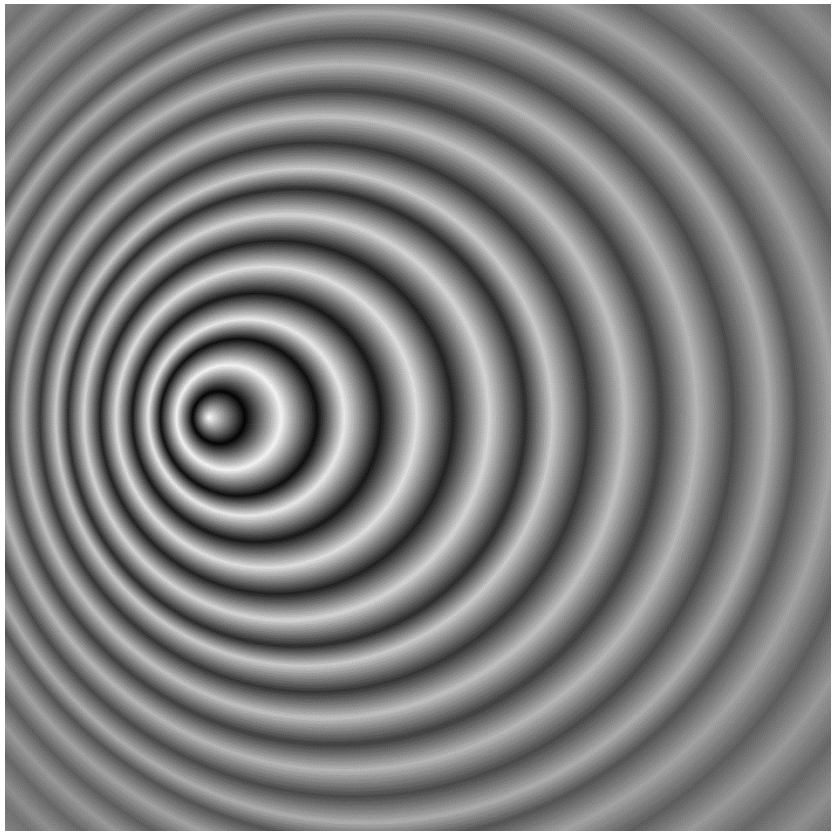
$$f_{\mathrm{B}} = f_{\mathrm{S}} \cdot \frac{c + v_{\mathrm{B}}}{c - v_{\mathrm{S}}}$$

$$f_{\mathrm{B}} = f_{\mathrm{S}} \cdot \frac{c - v_{\mathrm{B}}}{c + v_{\mathrm{S}}}$$

B: destination

S: source

c: speed of sound



Sound reflections

TECHNISCHE UNIVERSITÄT DARMSTADT

Highly dependent on surface structure and wavelength

Large surface, small wavelength

- Direct reflection

Rough surface

- Scatters sound

Effects

Echo, Reverb

- Direct reflections
- Replay sound with reduced amplitude

Damping

- Occluders

Calculating Reflections

TECHNISCHE UNIVERSITÄT DARMSTADT

Requires 3D model of the scene

- Geometry
- Surface properties

Kind of like 3D graphics

A3D

TECHNISCHE UNIVERSITÄT DARMSTADT

3D audio API from Aureal

Supports modelling and simulating 3D environments for sound

Only supported on special hardware

- From the late 90s

A3D today

TECHNISCHE UNIVERSITÄT DARMSTADT

Creative Labs bought Aureal in 2000

- After they lost a patent war in court

A3D functionality mostly integrated in EAX

EAX deprecated

EAX functionality integrated in OpenAL (EFX)

OpenAL kind of deprecated -> OpenAL Soft

Sound Effects and Music today

TECHNISCHE UNIVERSITÄT DARMSTADT

Mostly primitive

- Streaming prerecorded music
- Basic sound effects playback
  - Maybe some simple environmental effects

Scripting

TECHNISCHE UNIVERSITÄT DARMSTADT

Scripting language <-> Compiled language

JIT-compiled languages?

No clear definition

- Typically something easier than C++
  which is interpreted or compiles very fast

History

TECHNISCHE UNIVERSITÄT DARMSTADT

Games programmed in assembler

Especially annoying for story based games

- Lots of text
- Cut-scenes,…
  - Walk to x, y
  - Start dialog z
  - …

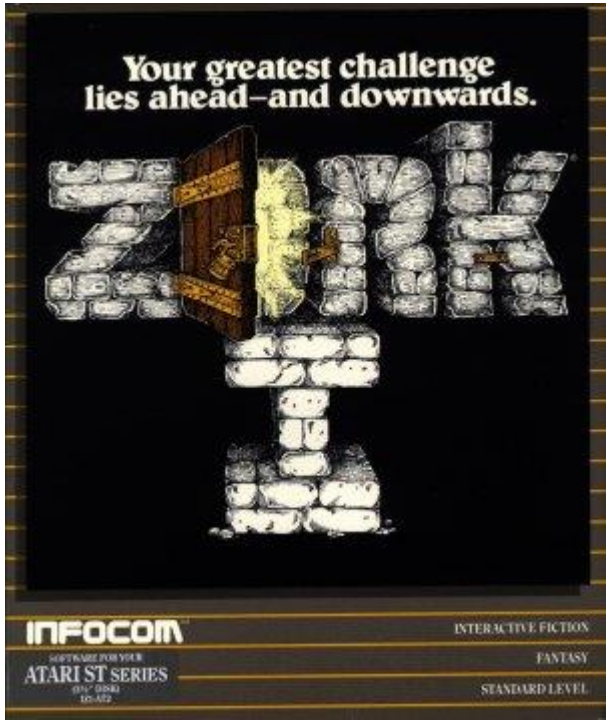
Hard to port games

History

Zork Implementation Language (ZIL)

- 1979
- Created by Infocom to facilitate the creation of interactive fiction titles
- Compiles to code for a virtual machine → Z-Machine

```
<ROOM LIVING-ROOM
      (LOC ROOMS)
      (DESC "Living Room")
      (EAST TO KITCHEN)
      (WEST TO STRANGE-PASSAGE IF CYCLOPS-FLED ELSE
           "The wooden door is nailed shut.")
      (DOWN PER TRAP-DOOR-EXIT)
      (ACTION LIVING ROOM-F)
      (FLAGS RLANDBIT ONBIT SACREDBIT)
      (GLOBAL STAIRS)
      (THINGS <> NAILS NAILS-PSEUDO)>
```



History

AGI – Adventure Game Interpreter

- 1984
- Created by Sierra On-Line for graphical adventure games
- First used fully in King‘s Quest
- Superseded by SCI – Sierra Creative Interpreter

```
if (said("look","door")) {
  if (posn(ego,0,120,159,167)) {
    print("These doors are strongly built
      to keep out unwanted visitors.");
  }
  else {
    print("You can't see them from
      here.");
  }
}
```

History

SCUMM – Script Creation Utility for Maniac Mansion

- 1987
- Created by Lucasfilm Games for… Maniac Mansion

```
cut-scene {
    ...
    actor nurse-edna in-room edna-bedroom at 60,20
    camera-follow nurse-edna
    actor nurse-edna walk-to 30,20
    wait-for-actor nurse-edna
    say-line nurse-edna "WHATS'S YOUR POINT ED!!!"
    wait-for-talking nurse-edna
    ...
}
```

History

TECHNISCHE UNIVERSITÄT DARMSTADT

Another World

Runs a fantasy machine code

Runs multiple „threads“ in parallel

- Cooperative multithreading
- Move multiple characters at the same time

Was ported to everything

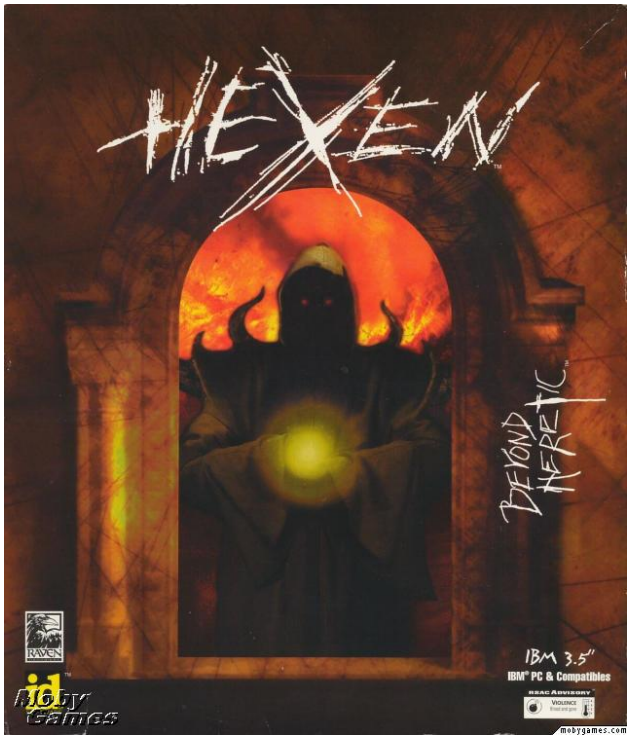
History

TECHNISCHE UNIVERSITÄT DARMSTADT

Action Code Script

- 1995
- Created by Raven Software for Hexen, extending the original Doom engine
- Allowed scripting events during a level

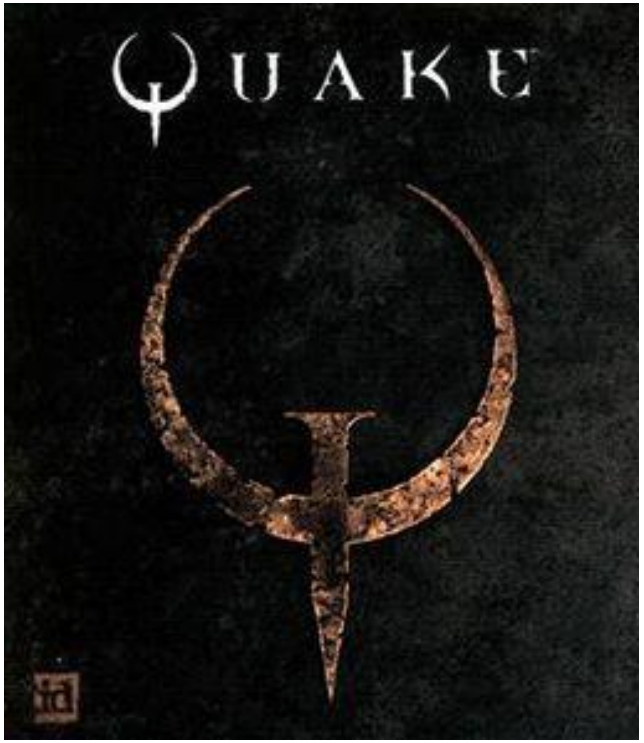
```
SCRIPT 4 (void)
{
  suspend;
  suspend; // The statements "absorb" the
effect of the two first toggles, whichever
they are
  ambientsound("Chat",127);
  printbold(s:"SEQUENCE COMPLETED!");
}
```



History

QuakeC

- 1996
- Created by id Software to control Quake




```
void (float v) ai_berserk =
{
  if (self.health > 25)
  {
    ai_run (v);
  }
  else
  {
    ai_run (v * 1.5); // adjust to your taste
    self.nextthink = time + 0.075; // adjust to your taste
  }
};
```

History

TECHNISCHE UNIVERSITÄT DARMSTADT

Unreal Engine 3

UnrealScript

- Developed in 1998 for the first Unreal

Can extend the class hierarchy of Unreal

```
class HappyHUD extends HUD
        config(Game);

//VARS
var String joyMessage;
var FONT joyFont;

…
function DrawHUD()
{

  local string StringMessage;

  //projection of ray hit location
  local vector HitLocation, HitNormal;

  //get origin and dir
  Canvas.DeProject(MousePosition, WorldOrigin, WorldDirection);

  StringMessage = "MouseX" @ MousePosition.X @ "MouseY" @ MousePosition.Y;
```

Unreal Engine 4

Unreal Script removed

- Programming in C++ plus macros
- Alternatively visual scripting

Advantages of Scripting Languages

TECHNISCHE UNIVERSITÄT DARMSTADT

Designer-Friendly

- Designers, non-programmers are enabled to work on the game directly, without needing programmer resources (ideally…)
- Quickly change values, …

Easy to learn

- Often reduced complexity compared to C++ or similar languages
- Often no memory management, pointers, …

Adaptable

- Many scripting languages are flexible
- E.g. Ruby or Lua allow adapting the language itself, e.g. to create a domain-specific language

Advantages of Scripting Languages

TECHNISCHE UNIVERSITÄT DARMSTADT

Concurrency

- Coroutines
- Functions that can be interrupted and continued

No compilation

- No additional time during compiling the game (engine)
- Can be switched during runtime
- Downside: Often slower than compiled code

Mod-support

- Allows players to change the game using the scripting language
- Increases shelf-life

Common language properties

TECHNISCHE UNIVERSITÄT DARMSTADT

(Seemingly) Interpreted

- Flexibility, portability and rapid iteration
- Virtual machine → port the VM to port the scripts

Lightweight

- Simple, low memory footprints

Support for rapid iteration

- Quicker turnaround time
- See changes immediately/after a restart

Convenience

- Tuned for the purpose in the game

Textual Languages

TECHNISCHE UNIVERSITÄT DARMSTADT

All languages we have seen so far

Special case: Natural-language Programming

Can be found in Inform 7 (interactive fiction tool)


```
The shower is here. It is fixed in place. "Opposite the
  mirror is the shower, which is closed." The description of
  the shower is "When it's open, you get in it to take a
  shower. Right now it's closed, keeping you from using it."

Instead of opening or entering the shower, say "It is locked
  down until after the ship makes its jump to hyperspace."
```

Textual Languages

TECHNISCHE UNIVERSITÄT DARMSTADT

All languages we have seen so far

Special case: Natural-language Programming

Can be found in Inform 7 (interactive fiction tool)


```
The shower is here. It is fixed in place. "Opposite the
  mirror is the shower, which is closed." The description of
  the shower is "When it's open, you get in it to take a
  shower. Right now it's closed, keeping you from using it."

Instead of opening or entering the shower, say "It is locked
  down until after the ship makes its jump to hyperspace."
```

Lua

TECHNISCHE UNIVERSITÄT DARMSTADT

http://www.lua.org/

Development started in 1993 at Pontifical Catholic University of Rio de Janeiro

Small language core

„Events“

- Fired when operators/functions are called, …
- Native code can register to handle them

Tags

- Code called when events are fired
- Allow Lua behaviour itself to be changed

Lua Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Used in Grim Fandango

http://www.lua.org/wshop05/Mogul.pdf

- – Dialogue
- – Puzzle logic
- – UI/controls
- – Menus
  - Engine handles only animations, backgrounds, sound, rendering, choreography, etc etc etc... But those aren't Grim Fandango

Python

TECHNISCHE UNIVERSITÄT DARMSTADT

https://www.python.org/

Development started in 1989 by Guido van Rossum as a hobby project

Easier to learn for non-programmers than other languages

Disadvantages: Large size and speed

- Relies on hash table lookups

Eve Online server almost completely written in Stackless Python

Visual Languages

TECHNISCHE UNIVERSITÄT DARMSTADT

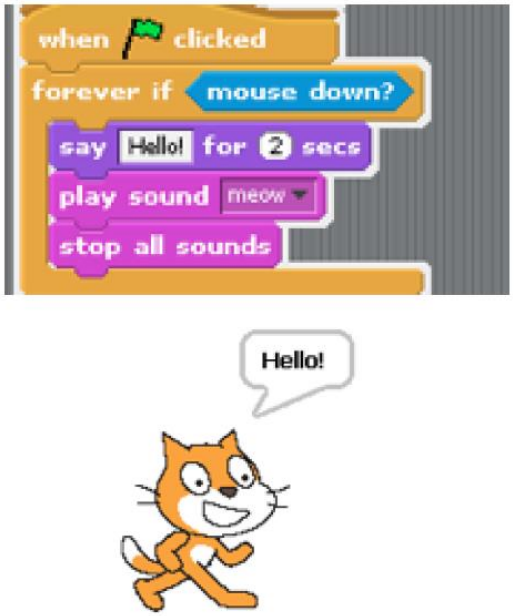
Sometimes trendy

Designer-friendly, easy to debug/visualize scripts

Can become complex easily

Visual Languages: Scratch, Storytelling Alice

TECHNISCHE UNIVERSITÄT DARMSTADT



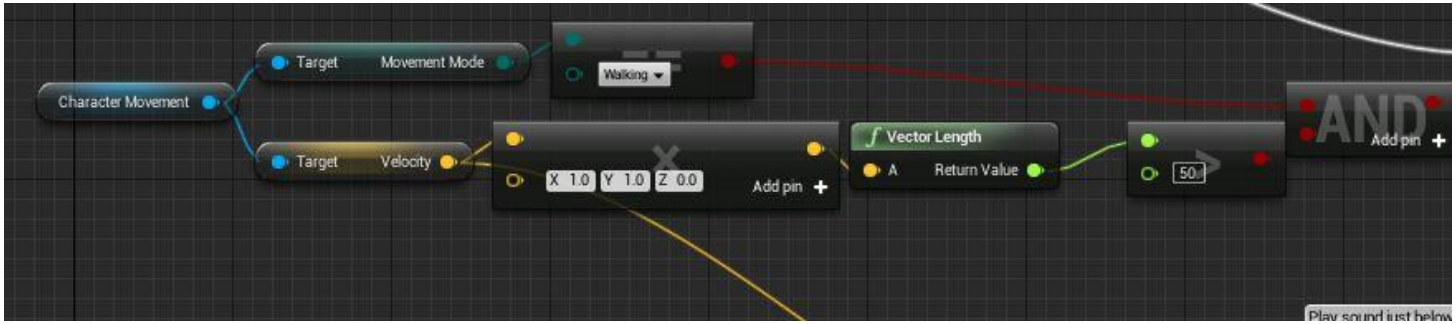
Visual Language: Unreal Blueprint

TECHNISCHE UNIVERSITÄT DARMSTADT

Added in Unreal Engine 4

Can extend C++ classes

Graph-based scripting language



Multithreaded scripts

TECHNISCHE UNIVERSITÄT DARMSTADT

Usually done via cooperative multitasking

Language support for coroutines

Scripts explicitly yield to other scripts

- Wait for x seconds
- Wait for x frames

Examples

- Can be realized in Lua
- Unity

Hot reload/patching

TECHNISCHE UNIVERSITÄT DARMSTADT

C++ is very restricted in changing code at runtime

- Compilation times
- No structural changes possible

Scripting languages can help

- Some can be modified freely at runtime

Very helpful to avoid reloading and replaying levels

Scripting trends?

TECHNISCHE UNIVERSITÄT DARMSTADT

id software does everything in C++ starting with Rage

- Only hires pro devs

What to script is very different in different games

All kinds of visual tools used to supplement scripting

Is Java a game engine?

TECHNISCHE UNIVERSITÄT DARMSTADT

Core in C++

Loads Java bytecode at runtime

Bytecode is executed, interacts with C++ components

Similar to Unreal Script or Mono in Unity

Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

Non-programmer friendly

- Designers can test/iterate quickly
- Mod support

Quick iteration times

- More simple to program than full programming language
- Hot reload

But: Not for everything

- Performance critical code
- Complex code