

Mit k(l)einem Budget Coding-Workshops über das Internet durchführen (ein Praxisbericht)

Bitte das Mikrofon nur dann aktivieren/einschalten,
wenn man selbst etwas sagen möchte!



Vorstellung

Thomas Ederer

Nachrichtenelektroniker, Informatiker,
„Orchideenvermehrer“

Softwareentwickler bei der Frequentis AG
Initiator des CoderDojo Neusiedl am See



thomas@neusiedl.coderdojo.net

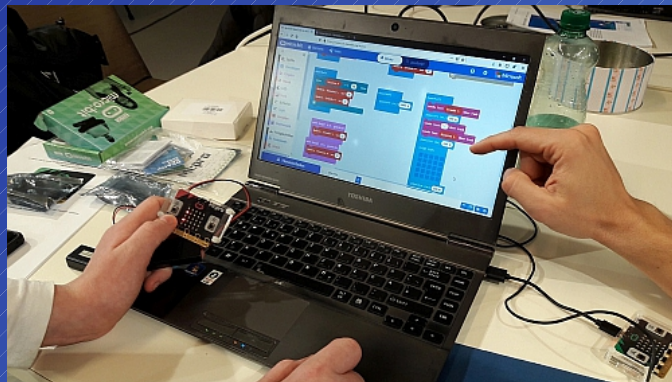
CoderDojo

2252 kostenlose Programmierklubs in 111 Ländern

12000 ehrenamtlich tätige Mentor_innen

5 CoderDojos in Österreich

- Linz
- Steyr
- Horn
- Wien
- Neusiedl am See
- Graz



<https://coderdojo.com>

Inhalt

- Technische Voraussetzungen
- Verwendete Plattformen
- Tools
- Formate
- Beispiele
- Erfahrungen
- Ideen für künftige Workshops

Voraussetzungen

Moderator_in

- Rechner mit Headset oder Freisprecheinrichtung
- zwei Bildschirme



Teilnehmer_in

- Rechner oder Tablet mit Mikrofon

Co-Moderator_in kann beim Bearbeiten der Chatnachrichten sehr hilfreich sein!

Plattformen

Microsoft Teams

zoom

BlueJeans

Hinweis: Die Clients belasten das System meist weniger stark als der Browser.

Tools

ZoomIt:



- Bildschirminhalt vergrößert anzeigen
- auf dem Bildschirminhalt zeichnen/schreiben
- ein Sketch Pad für schnelle Skizzen
- Speichern des Bildschirminhalts in die Zwischenablage oder in eine Datei
- Countdown

<https://docs.microsoft.com/en-us/sysinternals/downloads/zoomit>

Tools

Snagit:



- umfangreiche Unterstützung für das Erstellen von Videos (z.B. Tutorials), Arbeitsblättern, Grafiken uvm.
- leider nur kostenlose Testversion verfügbar

<https://www.techsmith.de/snagit.html>

Tools

CryptPad:



- diverse webbasierte Anwendungen für das verteilte Arbeiten (z.B. Code, Umfrage, Whiteboards, Tabellen...)
- verschlüsselte Kommunikation

<https://pads.c3w.at>

Tools

IrfanView:



- kompaktes Bildbearbeitungsprogramm
- einfache Bedienung

<https://www.irfanview.com>

Inkscape:

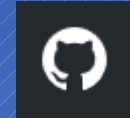


- Vektorgrafikeditor
- Export in Bitmaps (PNG)

<https://inkscape.org>

Tools

GitHub:



- Versionskontrollsystem
- gratis statische Webseiten (GitHub Pages)

<https://github.com>

Git:



- Client für jedes Betriebssystem
- auch als Portable verfügbar

<https://git-scm.com>

Tools

Notepad++:

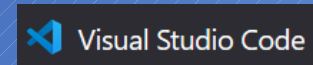
- kompakter Texteditor



<https://notepad-plus-plus.org>

Visual Studio Code:

- Texteditor
- Entwicklungsumgebung für viele Sprachen
- Auch für Linux verfügbar



<https://code.visualstudio.com>

IntelliJ IDEA:

- vollwertige Entwicklungsumgebung



<https://www.jetbrains.com/de-de/idea/>

Formate

- Eine Person macht die Übung vor – die Anderen machen es nach.
 - gut für Grundlagenschulungen
- Mob Programming - Die tippende Person wird in zuvor vereinbarten Intervallen (z.B. 10 Min.) gewechselt.
 - übt den Umgang mit Versionskontrollsystemen (z.B. Git)
 - hält die Teilnehmer_innen bei der Sache
 - trainiert Teamwork und Kommunikation

Beispiele

Java-Workshops

- Grundlagen
- Grafische Benutzeroberflächen mit Swing
- Spieleentwicklung

Beispiele

Java Grundlagen

- Unterschiede native Programme, virtuelle Maschinen mit ByteCode, interpretierter Code
- Nötige Schritte um aus einem Sourcecode ein lauffähiges Programm zu machen (Texeditor → javac → java)
- Datentypen, bedingte Anweisungen, Schleifen)
- unäre Zahlensystem

Beispiele

Konsolenausgabe:

Beispiele

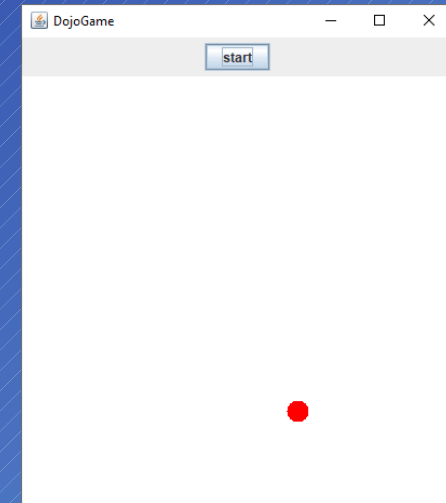
Grafische Benutzeroberflächen mit Swing

- Bestandteile eines Fensters
- Layout-Manager
- Icons
- ActionListener bei einem Button benutzen

Beispiele

Spieleentwicklung (erste Session)

- Grundlagen von Animationen
- interne Struktur von Spielen
- Funktion des Frameworks



Projekt: Ein Ball soll sich bewegen und am Bildschirmrand abprallen.

Lernziel: Punkte und Vektoren kennenlernen

<https://github.com/coderdojo-neusiedl/java-game-framework>

Beispiele

Spieleentwicklung (zweite Session)

- Figur mit Tastatur bewegen

Projekt: Vom oberen Rand des Spielfelds fallen Bälle nach unten, die eine Figur am unteren Spielfeldrand fangen soll.

Lernziel: Rechnen mit Vektoren.

<https://github.com/coderdojo-neusiedl/java-game-framework>

Beispiele

Spieleentwicklung (dritte Session)

- Bälle fallen lassen
- Kollisionserkennung einbauen

Projekt: Vom oberen Rand des Spielfelds fallen Bälle nach unten, die eine Figur am unteren Spielfeldrand fangen soll.

Lernziel: Abstand von Punkten berechnen.

<https://github.com/coderdojo-neusiedl/java-game-framework>

Beispiele

In weiteren Dojos eingesetzte Frameworks:



<https://scratch.mit.edu>



<https://microbit.org>



<https://www.tinkercad.com/learn/circuits>

JavaScript, C#, Webdesign, ...

Erfahrungen

- Getesteten Plattformen sind alle benutzbar.
 - Teams und BlueJeans fühlen sich recht ähnlich an und sind recht intuitiv benutzbar (geringe Einarbeitungszeit).
 - Zoom kann mehr, ist dadurch aber ein klein wenig intensiver bei der Einarbeitung.
- Wir konnten mehr Teilnehmer_innen gleichzeitig betreuen als es bei Anwesenheit möglich ist.
- Das Installieren von Software (Compiler, IDE, ...) bei den Teilnehmer_innen kann recht zeitraubend sein.
- Die technischen Probleme bei den Teilnehmer_innen bezüglich Videokonferenz hielten sich sehr in Grenzen

Ideen

- Workshop-Umgebung als Paket (z.B. Docker) oder virtuelle Maschine bereitstellen
- Workshops
 - Linux Know-how
 - 3D-Modelling mit Blender
 - OpenGL/WebGL
 - Snap! (<https://snap.berkeley.edu>)

Veranstaltungen

virtuelles CoderDojo (17.4.)

<https://mailchi.mp/dea860d74f2a/coderdojo-2020-04-17?fbclid=IwAR1ArhPaRMCGPzI54C2VHmvJjJWY8XIJu82MYIzcSL5hp-6Gow-VVlzKToM>

Global Azure Virtual Linz (24.4.)

<https://www.globalazurebootcamp.at>

Fragen?