

01

```
mirror_mod = modifier_ob.  
set mirror object to mirror  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

$(1+x+y+2a)-(3a+3g+x)$   
 $5+x+k+2a+21$   
 $E=mc^2$   
 $1+x+y+2a$   
 $1 \lim_{h \rightarrow 0}$   
 $2+\dots+2a+\dots+a$   
 $1+x+y+2a+21$   
 $1 \lim_{h \rightarrow 0}$   
 $x=0 \ x^n$   
 $\sum_{x=0}^n \{x-12-y+n\}$   
 $45-4a-3$   
 $\{x-12-y+n\}$

# Programmieren mit LiveCode

```
context):  
context.active_object is not
```



## Programmieren mit LiveCode

### Sekundarschule (7. - 9. Schuljahr)

### Schülerkurs

### Kompatibel mit dem Lehrplan 21:

#### Die Schülerinnen und Schüler können...

...verstehen, dass ein Computer nur **vordefinierte Anweisungen** ausführen kann und dass ein **Programm eine Abfolge von solchen Anweisungen** ist.

...können **selbstentdeckte Lösungswege** für einfache Probleme in Form von lauffähigen und korrekten Computerprogrammen mit **Schleifen**, **bedingten Anweisungen** und **Parametern** formulieren.

... können **selbstentwickelte Algorithmen** in Form von lauffähigen und korrekten Computerprogrammen mit **Variablen** und **Unterprogrammen** formulieren.

### LiveCode für die Schule?

... **kostenlos**

... **multiplattform** (Mac - Windows - Linux)

... **Scriptsprache dem Alltagsenglisch nahe**

... „**real Coding**“

... klassenunterricht für die 1. bis 3. Sek \*\*\*\*

... Unterricht im Rahmen des 9. Schuljahres / Projektarbeiten \*\*\*\*\*

... persönliche Fortbildung von Lehrpersonen \*\*\*\*\*

... Entwicklung von Tools für Lehrpersonen \*\*\*\*\*

\*\*\*\*\* sehr gut geeignet

\*\*\*\* gut geeignet

## **Inhaltsverzeichnis Schülerkurs**

1. Spesenbericht (Rechenblatt)
2. MyContacts (Adressdatenbank)
3. Mini\_Kino (Daumenkino)
4. Juke\_Box (Audio-Player)
5. Movie\_Player (FilmPlayer)
6. Memory
7. Space\_Pursuit (Game / 2 Spieler)

## **Ausgangslage**

*Im Zürcher Lehrplan 21 ist im Zyklus drei zu lesen:  
(<http://zhv.lehrplan.ch/index.php?code=a10|0|2|0|2>)*

### **Die Schülerinnen und Schüler können...**

...verstehen, dass ein Computer nur vordefinierte Anweisungen ausführen kann und dass ein Programm eine Abfolge von solchen Anweisungen ist.

...können selbstentdeckte Lösungswege für einfache Probleme in Form von lauffähigen und korrekten Computerprogrammen mit Schleifen, bedingten Anweisungen und Parametern formulieren.

... können selbstentwickelte Algorithmen in Form von lauffähigen und korrekten Computerprogrammen mit Variablen und Unterprogrammen formulieren.

Der vorliegende Schülerkurs deckt diese Anforderungen des LP 21 ab, indem die sachlichen Vorgaben behandelt (Variable, Schleifen ..... ) und die SUS befähigt werden, eigenständig einfache Probleme mit Hilfe selbst entwickelter Programme zu lösen.

### **Weshalb LiveCode für die Schule?**

Vorab ein wichtiges Argument: Die Community-Version ist kostenfrei, kann aber praktisch alles, was die Profiversion kann. Die Programmierung in LiveCode erfolgt in einer Scriptsprache, welche dem Alltagsenglisch sehr nahe ist. Zunächst werden die Programme in sogenannten Stacks auf der jeweiligen Plattform (MAC, Windows oder Linux) erstellt, der aus beliebig vielen Screens (sogenannten Karten) bestehen kann. Schliesslich kann dieser Stack als Runtimeversion für verschiedene Betriebssysteme kompiliert werden. Das heisst, die Stacks werden zu Programmen, die eigenständig, ohne Programmierumgebung, auf der jeweiligen Systemplattform laufen können. Gerade hier liegt eine der Stärken von LiveCode: Entwickeln auf der Systemplattform, an die ich mich gewöhnt bin, verteilen in Form von Runtimeversionen für die andern Plattformen. Ein Stack, der z.B. mit Mac begonnen wurde, kann problemlos mit Windows weiter bearbeiten und fertiggestellt werden: Entwickeln auf einer oder mehreren Plattformen

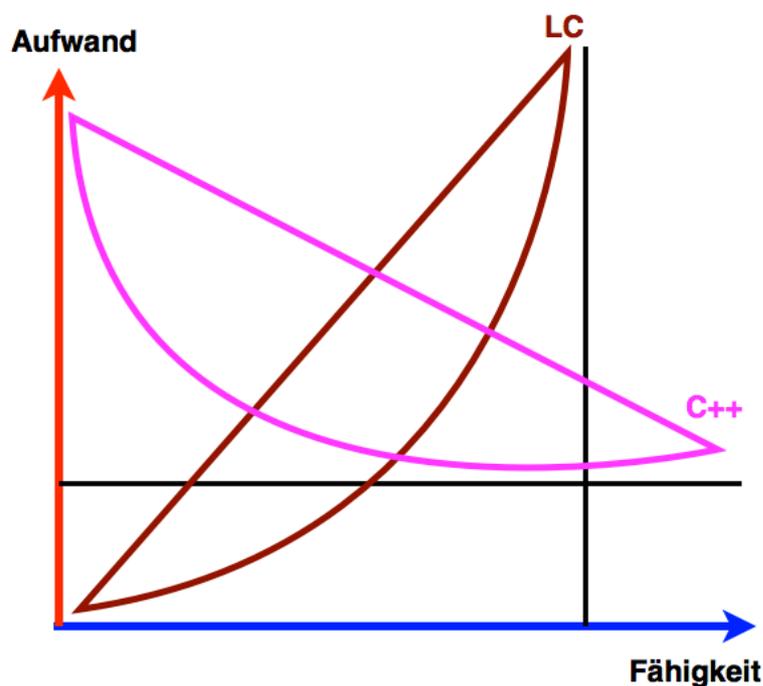
(Mac, Win, Linux), verteilen als Runtimeversion auf die andern Plattformen = Crossplattform-Prinzip. Das geht sogar so weit, dass aus einem auf einem Desktop entwickeltes Programm auch ein Programm für ein mobiles Gerät gemacht werden kann (iOS, Android).

Im Moment ist es so, dass die Bildungsdirektion des Kantons Zürich Scratch für den Einsatz in der Schule favorisiert. Ich finde hier, dass für einen ersten Einstieg in der Primarschule in die Welt des Programmierens Scratch eine gute Lösung ist. Für anspruchsvollere Programmierungen wird es mit Scratch aber schnell schwierig, Lösungen zu finden, wenn man nicht genau weiss, wie Scratch auszureizen ist: Sehr schnelle Lernkurve, aber schon bald sind Grenzen erreicht. LiveCode geht weiter: Auch hier sind schnelle Lernfortschritte möglich, die Grenzen nach oben werden aber erst viel später erreicht - so spät, dass auch professionelle Programmierer LiveCode als RapidApplicationDevelopmentTool (RADT) in der produktiven Arbeit einsetzen. Natürlich kann man auch sagen, dass LiveCode seine Grenzen hat und dass es doch besser wäre, gleich Objective C zu lernen. Das ist richtig, wenn man eine Lehre als Applikationsentwickler macht und eine ganze Lehre die Zeit dafür hat, diese äusserst anspruchsvolle Art der Programmierung zu lernen und nicht auf schnelle Fortschritte angewiesen ist. Dafür ist dann natürlich die Grenze nach oben wirklich offen.

Hier meine ich, dass wir getrost mit den Schulen in den USA, Schottland und diversen Unis in verschiedenen Ländern parallel laufen und LiveCode auf der Sekundarschule anwenden können. LiveCode eignet sich im Informatikunterricht für die Lösung einfacher Programmierprobleme oder für einen Freifachkurs im Rahmen des 9. Schuljahres in Verbindung mit einem entsprechenden Profil.

Der vorliegende Kurs eignet sich für das Selbststudium von Lehrpersonen oder aber eben für einen Freifachkurs im Rahmen des 9. Schuljahres in Verbindung mit einem entsprechenden Profil.

Vergleich von LiveCode mit Scratch und „grossen“ Programmiersprachen



	<b>Scratch</b>	<b>LiveCode</b>	<b>Objective C</b>
Auslegung	objektorientiert	objektorientiert	teils objektorientiert
Programmiersprache	hauptsächlich Anordnung von Objekten per Drag and Drop	dem Alltagsenglisch sehr nahe liegende Scriptsprache (Sprache / Syntax)	sehr komplexe / abstrakte Programmiersprache
Lernkurve	sehr schnell sind ansprechende Resultate möglich	grosse Fortschritte in kurzer Zeit, daher schnell ansprechende Resultate	grosser Lernaufwand für zu Beginn bescheidene Resultate
Möglichkeiten	im Vergleich mit LiveCode und Objective C schnell eingeschränkt	im Vergleich mit Objective C nach der Erreichung eines beachtlichen Levels immer mehr Einschränkungen	nach lange andauerndem Lernaufwand nach oben offen, praktisch ohne Grenzen nach oben
Plattformen	Windows, Linux, Mac Os	Windows, Linux, Mac Os und weitere Plattformen. Auch iOS für iPad, iPhone; ebenso Android.	Hauptsächlich Mac OS und iOS
Zielpublikum	Kinder, Jugendliche, Lehrpersonen „educational“. „Einfach“, da mit Drag and Drop gearbeitet wird. Kein „Coding“ im Sinne von „Programmieren“.	Jugendliche, Erwachsene, Lehrpersonen, Pros und Amateure. Eher nicht für Primarschule geeignet, hingegen für Sekundarschule (vor allem im 9. Sj. im Rahmen des Profilunterrichtes). „Real Coding“ im Sinne von „Programmieren“ innerhalb der vorgegebenen, relativ einfachen und verständlichen Programmiersprache.	eher für Pros, da sehr abstrakt / komplex. „Real Coding“ im Sinne von „Programmieren“ innerhalb der vorgegebenen, abstrakten und komplexen Programmiersprache.
Kosten	Keine Kosten, Freeware	Im Rahmen der Community Version, die alles kann, kostenlos.	Keine Kosten, Freeware

## **Aufbau des Schülerkurses / Hilfestellung**

Der Kurs ist methodisch so aufgebaut, dass die SUS im Projekt 1 bezüglich Programmierung moderate Anforderungen erfüllen müssen; die Hilfestellung vom Kurspapier her ist hingegen hoch. Es geht im Projekt 1 vor allem darum, die Programmierumgebung grob kennen zu lernen. Jedes weitere Projekt setzt das Erlernte aus dem vorangehenden voraus und baut zum Teil auch darauf auf. Im neuen Projekt wird Neues gelernt, die Anforderungen steigen, während die Hilfestellung sinkt. So soll erreicht werden, dass die SUS fortlaufend kompetenter, aber auch selbständiger werden.

- Im Projekt 1 kleine Anforderung, aber grosse Hilfestellung (SUS sollen Programmierumgebung kennen lernen) / im Projekt 7 grosse Anforderungen, kleine Hilfestellung.
- Die SUS nehmen das Gelernte von jedem vorhergehenden Projekt zum nächsten mit und werden bei steigenden Anforderungen und kleiner werdender Hilfestellung immer selbständiger.

## **Teile des Kurses**

1. PDF des Schülerkurses (100 Seiten)
2. Materialien für die Lehrperson (Data\_Teacher), beinhaltet sämtliche funktionsfähigen Programme als funktionierender Stack und als Standalone für Mac und Windows.
3. Materialien für die SUS (sämtliche Materialien, auf die im Schülerkurs verwiesen, bzw. im Schülerkurs verwendet werden).

## **LiveCode-Kurs beim ZAL**

Der ZAL führt meinen LiveCodekurs im offiziellen Kursangebot.

Nach spezieller Abmachung sind auch von mir geführte Kurse im Rahmen einer internen Lehrerfortbildung vor Ort in Schulhäusern möglich.

## **Links**

Die aktuellen Downloads von LiveCode

<http://downloads.livecode.com/livecode/>

Der vorliegende Kurs basiert auf der Version 8.15. Die Community Version ist kostenfrei, kann aber einen grossen Teil der professionellen Version.

Martin Meili  
[bmeili@swissonline.ch](mailto:bmeili@swissonline.ch)