



Automatisierung

Let's Play Automatisierung

📅 26. Juni 2017 (<http://sebinside.de/2017/06/26/lets-play-automatisierung/>) 👤 seb
(<http://sebinside.de/author/seb/>) ➔ Coding (<http://sebinside.de/category/coding/>)

Videoschnitt ist spannend und sehr kreativ – keine Frage. Gerade Formate wie VLOGs oder Reviews leben vom aufwendigen Schnitt und vom Rumprobieren mit neuen Techniken. Anders sieht das allerdings bei Let's Plays aus: Hier sind oft immer wieder gleiche Arbeitsschritte notwendig. Das Resultat ist kein kreativer Prozess, sondern Fließbandarbeit. Das nervt und frustriert. Deswegen stelle ich mir schon seit einiger Zeit die Frage: Kann man das nicht vernünftig automatisieren?

Die Vorgeschichte

Die Let's Play Szene ist auf YouTube irgendwann zwischen 2008 und 2010 entstanden. Damals waren Online-Gaming-Videos noch Neuland, die bekanntesten Gaming-YouTuber hatten nicht mal 1000 Abonnenten. Auch von Livestreaming hat noch niemand gesprochen. Dementsprechend klein war auch das Software-Angebot.

Wollte man qualitativ hochwertig produzieren, gab es da genau ein Programm: FRAPS. Riesige Output-Dateien, schlechte Performance... aber das Beste auf dem Markt.

Gleichzeitig war aber auch die Anforderung an Let's Plays entsprechend gering. Keine Schnitte, keine Intros. Ja, noch nicht einmal eine Begrüßung wurde gefordert. Zu dieser Zeit war es absolut ausreichend, zwei Stunden am Stück Ton und Video aufzunehmen, das ganze irgendwie zusammen zu rendern, in kleine 10-Minuten-Stücke zu teilen (YouTube hatte damals tatsächlich ein solches Limit für die Video-Länge) und hochzuladen.

Seitdem hat sich unglaublich viel verändert. Ohne jede Entwicklung im Detail zu erklären, reicht schon ein Blick auf die Reichweite der Gaming-Community in Deutschland (mehrere Millionen Zuschauer) oder die Anzahl der angebotenen Softwaretools (mehrere Dutzend), um zu erklären, was hier inzwischen los ist. Mittlerweile wird sogar extra Streaming-Hardware produziert, wie Gaming-Capture-Karten oder das Stream Deck von Elgato (<http://amzn.to/2tetvW2>).

Gleichzeitig sind aber natürlich auch die Anforderungen extrem gestiegen. Jede Folge braucht ihren eigenen Inhalt, eine Begrüßung, ein Intro, coole Übergänge, Schnitte oder Zusatzeffekte wie Zeitraffer. Zusätzlich wird natürlich eine Facecam eingefügt, diese wie auch die Stimme nachbearbeitet und Hintergrundmusik darf auch nicht fehlen. Zeitlich gesehen bedeutet dieser Zusatzaufwand, dass für ein 10-15 Minuten Video ohne Optimierung des Workflows gerne mal 30 Minuten bis mehrere Stunden Schnittaufwand dazu kommen. Täglich, und für jedes Video. Eine lineare, langweilige und unglaublich frustrierende Zeitverschwendung.

Die legendäre Start-Stopp-Aufnahme

Nachbearbeitung ist nichts anderes als Aufbereitung von bereits vorhandenem Inhalt. Dieser ändert sich jedoch nicht grundlegend, sondern wird nur „schön angemalt“. Aus diesem Grund haben schon vor mehreren Jahren Let's Player damit

begonnen, auf jegliche Schnitte bzw. Nachbearbeitung zu verzichten. Der Schlüssel hierzu sind Streaming-Tools wie xSplit (<https://www.xsplit.com/de/>) (kostenpflichtig) oder Open Broadcaster (<https://obsproject.com/>) (kostenlos und Open-Source).

Die zugrundeliegende Technik ist schnell erklärt: Da es sich hier eigentlich um Livestreaming-Software handelt, wird in Echtzeit die Stimme mit dem Videomaterial kombiniert und gerendert. Das Endprodukt sind fertige Videodateien, die so direkt hochgeladen werden können. Der Aufwand ist sehr gering, leider stark auf Kosten der Qualität. Dennoch sind Start-Stopp-Aufnahmen sehr beliebt und so effizient, dass diese schon wieder einen schlechten Ruf haben; von „liebloser Content-Massenproduktion“ ist oft die Rede.

Einfache Automatisierung

Hat man denn überhaupt keine Möglichkeit, beide Welten zu verbinden? Ein schneller, möglichst automatisierter Workflow, der an Start-Stopp-Aufnahmen herankommt, kombiniert mit guter Qualität? Doch, und ich war einer der ersten deutschen YouTuber, der sich mit dieser Thematik eingehend befasst hat. Macht man einige (kleine) Abstriche, kann man unglaublich viel im Rahmen der Echtzeit-Produktion verbessern.

Ein gutes Beispiel ist die Stimme: Auf jedem Livekonzert werden Stimme und Instrumente auch in Echtzeit optimiert. Warum ist das nicht bei Let's Plays möglich? Ich erinnere mich zurück an ein Gespräch, dass ich vor vielen Jahren (ich glaube es war 2012 oder 2013) mit dem YouTuber Dner (<https://www.youtube.com/felixladen>) hatte. Wir hatten zu diesem Zeitpunkt beide bereits unsere Stimmnachbearbeitung optimiert. Während Felix zusätzliche Hardware gekauft und zwischen Mikro und Sound-Interface eingesetzt hatte, habe ich alles virtuell gelöst. Sein Vorteil: Weniger Einrichtungsarbeit, mein Vorteil: Komplette kostenlos. Ohne ins Detail zu gehen: Es gibt da draußen verdammt gute kostenlose Plugins zur Klangverbesserung, die sich ohne Probleme in Echtzeit kombinieren lassen.

Aber bei der Stimme hört die Optimierung noch nicht auf. Spielt man während der Aufnahme im Hintergrund leise Musik ab, wird diese mit aufgenommen und man spart sich einen weiteren Schritt in der Nachbearbeitung. Fügt man während der Aufnahme im Streaming-Tool bereits die Facecam klein im Eck ein, muss diese ebenfalls nicht separat verarbeitet und gerendert werden. Allerdings steht dann natürlich auch kein Gameplay ohne besagte Facecam zur Verfügung, da bereits in diesem Format aufgenommen wurde. Solche Abwägungen werden wir noch öfters sehen: **Flexibilität im Schnitt vs. Effizienz.**

Kombiniert man diese Schritte, macht die Aufnahme schon einen besseren Eindruck, sieht aber immer noch stark nach Start-Stopp-Aufnahme aus. Aus diesem Grund habe ich irgendwann angefangen, auch mein Intro live in die Aufnahme hinein zu rendern. Der Trick hierbei: Manche Video-Codecs erlauben die Speicherung von Transparenz-Informationen. So lassen sich auch coole Übergänge realisieren.

Mit dem Endprodukt dieser Workflow-Verbesserung habe ich über ein Jahr lang meine Videos produziert. Nur so war es im Zeitraum von Craft Attack 3 möglich, bis zu 4 Videos täglich hochzuladen. Was ist aber, wenn einem selbst diese Qualität nicht ausreicht?

Verbesserter Schnitt-Workflow

Niemand sagt, dass sich ein verbesserter Aufnahme-Workflow nicht auch mit weiterer Nachbearbeitung verträgt. Mein aktuelles Vorgehen ist es deshalb, bereits in der Aufnahme so viel wie möglich zu optimieren und nur die notwendigen Schritte im Nachhinein zu ergänzen.

Schauen wir uns das ganze anhand einer Folge FTB Skyblock (https://www.youtube.com/watch?v=3EZsxdX_dIM) an: Die Nachbearbeitung der Stimme als auch die Farbkorrektur der Facecam passieren live und in Echtzeit. Ebenfalls wird die Facecam wie oben beschrieben bereits zur Aufnahmezeit mit dem Gameplay verbunden. Auch die Übergänge (inkl. Soundeffekt) werden live und per

Tastendruck umgesetzt. Zusätzlich werden live auf einer weiteren Audiospur Schnittmarken gesetzt, wenn etwas Besonderes in der Aufnahme passiert, was einen Schnitt erfordern würde.

Im Nachhinein wird das im MKV-Format aufgenommene Material als MP4-Datei remuxed und in eine speziell vorgefertigte Premiere-Session importiert. Hier liegen bereits verschiedene Intros sowie vorgepegelte Musik bereit. Alle Elemente werden kombiniert, ggf. weitere Schnitte und Zeitraffer gesetzt, fertig. Mit dieser Vorgehensweise konnte ich meinen Zeitaufwand für den Schnitt von 20 Minuten auf 5 Minuten pro Folge reduzieren. Ebenfalls sind natürlich komplexere Optimierungen, wie Schnitte im Gameplay („Jump-Cuts“) und vieles mehr, denkbar.

Obgleich dieser Workflow schon weit optimiert ist, ist trotz allem noch zusätzliche Nachbearbeitung notwendig. Grundsätzlich wäre das kein Problem, wenn es nicht immer und immer wieder die gleichen Schritte wären. Und wie wir bereits am Anfang gelernt haben: Das nervt und könnte automatisiert werden!

Vollautomatisierung

Jetzt kommt der „coole Inhalt“. Die Techniken und Vorgehensweisen, die ich in diesem Abschnitt erkläre, wurden meines Wissens noch nie in der Let's Play Szene umgesetzt. Und das, obwohl vollautomatisierte Videoproduktion nichts Neues ist. Durch Zufall habe ich vor kurzem zwei andere Informatik-Master-Studenten getroffen, die sich an unserer Uni um die Aufnahme und Produktion der Vorlesungen kümmern. Eine fast vollständig automatisierte Pipeline, wie ich später erfahren habe.

Die grundlegende Idee ist einfach: Beschränkt man sich auf reine Schnitte; das heißt, man ändert nur die Reihenfolge von Videomaterial und fügt keine Effekte hinzu, kann man dies ohne Neu-Enkodierung tun. Klingt unspektakulär, hat allerdings einen riesigen Vorteil: Während das Rendern eines 15-Minuten-Let's Plays abhängig von der Nachbearbeitung mindestens mal 10-20 Minuten dauert, braucht das Neuankordnen von Video-Teilen nur wenige Sekunden Verarbeitungszeit. Außerdem lässt sich dies mit CLI-Tools wie FFMPEG (<https://ffmpeg.org/>) oder Avidemux

(<http://fixounet.free.fr/avidemux/>) ohne Kosten vollständig automatisieren. Das einzige, was nicht so einfach geht, ist der Audio-Teil. Hier geht die Encodierung aber glücklicherweise sehr viel schneller als bei Video-Dateien.

Um eine **Vollautomatisierung** zu erreichen, sind also die folgenden Schritte notwendig:

1. Alle Effekte (wie Facecam, Transitions usw.) müssen bereits zur Aufnahmezeit in das Video gerendert werden (Stichwort: Open Broadcaster)
2. Der Let's Player erstellt während der Aufnahme Schnittmarkierungen per Shortcut, um z.B. einen Offscreen-Part zu verwerfen oder einen Zeitraffer einzuleiten
3. Das Quellvideo wird auf Basis der Markierungen automatisiert zugeschnitten und mit vorhandenen Assets (wie z.B. Musik oder einem fertigen Outro-Screen) kombiniert

Natürlich ist dieses Vorgehen nur umsetzbar, wenn das Let's Play einem festen Format folgt (was es üblicherweise aber tut) und der Let's Player den Zusatzaufwand der Marker-Erstellung zur Aufnahmezeit in Kauf nimmt. Der Gewinn ist allerdings entsprechend hoch: Videoschnitt in wenigen Sekunden ohne einen Finger zu rühren. Quasi die intelligente Version einer Start-Stopp-Aufnahme. Und niemand sagt, dass das Ergebnis nicht doch noch im Schnittprogramm weiter verarbeitet werden kann: Da nichts neu enkodiert wird, hat das Endprodukt immer noch die Qualität der Originalaufnahme.

Realisierbarkeit

Obwohl mir nicht bekannt ist, dass diese Technik bereits in der Let's Play Szene eingesetzt wird, spricht aktuell nichts gegen eine Realisierung. Alle benötigten Techniken und Tools sind bereits vorhanden und sogar kostenlos.

Grundsätzlich benötigt man ein Aufnahmeprogramm (OBS, bereits vorhanden), ein PlugIn zum Erstellen von mit der Aufnahme synchronisierten Markern (OBSInfoWriter (<https://github.com/partouf/OBSInfoWriter>), bereits vorhanden. Danke für den Tipp

an Benjamin) und ein Tool zum Splitten, Mergen und muxen von Videodateien (FFMPEG, bereits vorhanden).

Der Knackpunkt ist nur die Kombination dieser Tools: Hierfür fehlt aktuell noch die Software. Natürlich ist das Hardcoden für ein Format kein großes Problem und ließe sich problemlos an einem Wochenende umsetzen. Will man allerdings Flexibilität und vor allem Ausfallsicherheit, am besten noch kombiniert mit einer intuitiven und schnell verwendbaren Oberfläche, wird alles sehr viel komplexer und teurer. Klingt aber auf jeden Fall nach einem vielversprechenden Projekt, das sicher nicht nur für mich, sondern auch für viele andere Let's Player relevant wäre...

Alternative Ansätze

Natürlich haben wir im Rahmen unseres 1,5-stündigen-Anti-Vorlesungs-Brainstormings auch andere Vorgehensweisen besprochen. Eine weitere Möglichkeit wäre die Einbindung einer Schnittsoftware wie Adobe Premiere. Mit Hilfe des Final Cut Interchange Formats (<https://documentation.apple.com/en/finalcutpro/usermanual/index.html#chapter=97%26section=3%26tasks=true>) ist es möglich, Änderungen an vorhandenen Schnittprojekten vorzunehmen, basierend auf diesen Daten andere Software zu steuern oder komplette Projekte aus dem Nichts zu generieren.

Die Umsetzung hiervon ist auf Grund der Komplexität eines Schnittprojekts allerdings recht teuer. Einfache Schnittmarker lassen sich schnell auslesen, aber ein komplettes Projekt zu erstellen ist ein ganz anderes Ding. Und ich weiß, von was ich rede... ich habe testweise eine komplette Projekt-XML-Datei von Hand geschrieben.

Dieser Ansatz gefällt mir allerdings noch aus einem anderen Grund nicht: Er erfordert in irgendeiner Art wieder die Interaktion des Nutzers. Warum sollte man sich während der Aufnahme den Stress machen, Marker zu setzen, wenn man am Schluss trotzdem wieder an den Schnitt muss? Entweder vollautomatisch oder gar nicht.

Eine weitere Möglichkeit ist der Einsatz von neuronalen Netzen und maschinellem Lernen. Zugegeben: Das ist schon ziemlich abgefahren. Wir trainieren einen Computer mit gut geschnittenen Videos, er lernt daraus und schneidet uns dann unsere Videos, besser, als es ein Mensch kann? Klingt verrückt, wurde aber so ähnlich tatsächlich schon umgesetzt. Das Ganze sollte man natürlich nicht mit simpler Mustererkennung auf der Audiospur verwechseln, wie es es ein aktuelles Kickstarter-Projekt (https://www.kickstarter.com/projects/davidpennington/movieslice-an-automatic-video-editor?ref=category_popular) vormacht.

Fazit

„Alles, was du mehr als einmal machst, hättest du auch automatisieren können“. Das ist ein schönes Zitat, um diesen Post zu beenden, oder? Unsere Zeit ist knapp. Meine Zeit ist knapp. Ich würde sie lieber mit kreativer Arbeit füllen, anstatt die immer wieder gleichen Schritte zu wiederholen. Nochmal: Videoschnitt, z.B. bei VLOGs ist unglaublich spannend, kreativ und abwechslungsreich. Simple Fließband Let's Plays sind allerdings genau das Gegenteil, und wollen meist auch gar nicht mehr sein.

Natürlich erstellen viele Let's Player auch „Unikat-Videos“, die keinem festen Konzept folgen und nicht nach Schema-F geschnitten werden können. Meiner Ansicht nach kann man mit den vorgestellten Techniken aber auch hier viel optimieren. Letztlich geht es um die Einsparung von Zeit, die man viel besser anderweitig verwenden kann.

Hoffentlich finde ich irgendwann die Zeit, ein solches Tool testweise zu entwickeln. Sicher wäre dies auch für viele andere YouTuber relevant und abgesehen davon eine tolle Kombination vorhandener Technologien. Mal sehen, was die Zukunft bringt. Vielen Dank fürs Lesen, ich freue mich auf euer Feedback!

Teilen mit:

 E-Mail (<http://sebinside.de/2017/06/26/lets-play-automatisierung/?share=email&nb=1>)

 Facebook 1 (<http://sebinside.de/2017/06/26/lets-play-automatisierung/?share=facebook&nb=1>)

 Twitter (<http://sebinside.de/2017/06/26/lets-play-automatisierung/?share=twitter&nb=1>)

Gefällt mir:

Lade ...

[automatisierung](http://sebinside.de/tag/automatisierung/) (<http://sebinside.de/tag/automatisierung/>) [avidemux](http://sebinside.de/tag/avidemux/) (<http://sebinside.de/tag/avidemux/>)

[ffmpeg](http://sebinside.de/tag/ffmpeg/) (<http://sebinside.de/tag/ffmpeg/>) [let's play](http://sebinside.de/tag/lets-play/) (<http://sebinside.de/tag/lets-play/>)

[obs](http://sebinside.de/tag/obs/) (<http://sebinside.de/tag/obs/>) [open broadcaster](http://sebinside.de/tag/open-broadcaster/) (<http://sebinside.de/tag/open-broadcaster/>)

[optimierung](http://sebinside.de/tag/optimierung/) (<http://sebinside.de/tag/optimierung/>) [skate702](http://sebinside.de/tag/skate702/) (<http://sebinside.de/tag/skate702/>)

[youtube](http://sebinside.de/tag/youtube/) (<http://sebinside.de/tag/youtube/>)

7 Gedanken zu „Let's Play Automatisierung“

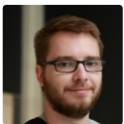


GERD2002

26. Juni 2017 um 20:19 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-113>)

Wenn du das machen solltest, dann bitte Open-Source, damit alle mithelfen können.

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=113#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=113#respond))



FELIX BEUSTER ([HTTPS://BEUSTERSE.DE](https://beusterse.de))

26. Juni 2017 um 20:37 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-114>)

Sehr spannendes Thema. 😊

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=114#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=114#respond))

7HOCHO

26. Juni 2017 um 20:47 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-115>)

Ich habe mich vor kurzem auch mit der Möglichkeit beschäftigt, Let's Plays vollständig zu automatisieren. Das einzige Problem, das ich bei deiner Idee gefunden habe ist, dass die Musik bei Zeitraffern auch schneller werden würde. Sonst klingt es sehr gut. Wahrscheinlich werde ich mir das mit dem Codieren von Videomaterial auch einmal anschauen. Vielleicht ist es sogar während der Aufnahme möglich, in OBS Timelapses hinzuzufügen und das Video zusammenzuschneiden.

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=115#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=115#respond))

KARL ([HTTPS://TWITTER.COM/KARLTHEBEE](https://twitter.com/karlthebee))

27. Juni 2017 um 17:07 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-116>)

Ich habe selbst schon mit dem OBS das ganze halbautomatisiert. Also Intro-Sequenz, eine andere für den Inhalt und eine Outro-Sequenz. Dazu mit einem kleinen Java-Programm eine variable Buachbinde erstellt.

Jedoch weiß ich nicht, wie gut die vollautomatisierung klappt : Also in wie fern man nebenbei redaktionell „Sprungmarken“ vormerken kann, ohne dass es störend oder ablenkend für den Zuschauer ist.

Oder beim machine learning finde ich es ziemlich schwer, tolle Szenen zu erkennen, wenn man nicht stupide nach lauten Geräuschen (lachen, schreien), was auf irgendwas interessantes hinweisen könnte, sucht.

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=116#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=116#respond))

LEXGASHD ([HTTPS://PLUS.GOOGLE.COM/109272798183588876333](https://plus.google.com/109272798183588876333))

29. Juni 2017 um 7:02 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-119>)

Wenn so was in die Realität um gesetzt würden wäre dann könnte sich die ganze lets play und Blog scene auf der Welt ändern

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=119#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=119#respond))

ROBERT ([HTTPS://PLUS.GOOGLE.COM/109694659279643637851](https://plus.google.com/109694659279643637851))

29. Juni 2017 um 7:59 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-120>)

Kann das ganze auch mit schlechterer Hardware noch gut funktionieren?

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=120#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=120#respond))

MINECURSER

13. November 2017 um 14:22 Uhr (<http://sebinside.de/2017/06/26/lets-play-automatisierung/#comment-127>)

Hey Sebastian,

könntest du evtl. deine kostenlosen „Soundqualität-Verbesserungs-Plugins“ einfach aufzählen. Würde mich da gerne selbst reinfuchsen. Wäre nett wenn ich schon mal wissen würde, mit welchen Plugins es sich gut arbeitet lässt.

Vielen Dank schon mal 😊

ANTWORTEN ([HTTP://SEBINSIDE.DE/2017/06/26/LETS-PLAY-AUTOMATISIERUNG/?REPLYTOCOM=127#RESPOND](http://sebinside.de/2017/06/26/lets-play-automatisierung/?replytocom=127#respond))

KOMMENTAR VERFASSEN

Gib hier deinen Kommentar ein ...

◀ Ein Blick auf die Modpack-Entwicklung (<http://sebinside.de/2017/05/08/ein-blick-auf-die-modpack-entwicklung/>)

Eine neue Plugin-Architektur für Code Overflow ▶ (<http://sebinside.de/2018/01/16/eine-neue-plugin-architektur-fuer-code-overflow/>)

SOCIAL MEDIA



(<http://twitter.com/sebinside>) (<http://facebook.com/sebinside>) (<http://youtube.com/sebinside>) (<http://github.com/sebinside>)

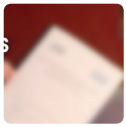
BLOG VIA E-MAIL ABONNIEREN

Gib Deine E-Mail-Adresse an, um diesen Blog zu abonnieren und Benachrichtigungen über neue Beiträge via E-Mail zu erhalten.

E-Mail-Adresse

ABONNIEREN

BELIEBTE BEITRÄGE



Rückblick auf den Bachelor (<http://sebinside.de/2017/04/13/rueckblick-auf-den-bachelor/>)

13 Apr , 2017



Programmierparadigmen

(<http://sebinside.de/2016/07/02/programmierparadigmen/>)

02 Jul , 2016



Webseiten kommen und gehen (<http://sebinside.de/2016/06/15/webseiten-kommen-und-gehen/>)

15 Jun , 2016

KATEGORIEN

📁 Allgemein (<http://sebinside.de/category/allgemein/>)

📁 Coding (<http://sebinside.de/category/coding/>)

📁 Minecraft (<http://sebinside.de/category/minecraft/>)

📁 Studium (<http://sebinside.de/category/studium/>)

Suche ...



NEUESTE KOMMENTARE

💬 Bowser bei Hallo. äh.. Welt! (<http://sebinside.de/2016/03/22/hallo-ae-h-welt/#comment-138>)

💬 daeaevaeaed bei Rückblick auf den Bachelor (<http://sebinside.de/2017/04/13/rueckblick-auf-den-bachelor/#comment-137>)

💬 coernerbrot bei Programmierparadigmen
(<http://sebinside.de/2016/07/02/programmierparadigmen/#comment-136>)

💬 seb (<http://sebinside.de>) bei Eine neue Plugin-Architektur für Code Overflow
(<http://sebinside.de/2018/01/16/eine-neue-plugin-architektur-fuer-code-overflow/#comment-135>)

💬 Sven (<https://usacookie.de>) bei Rückblick auf den Bachelor
(<http://sebinside.de/2017/04/13/rueckblick-auf-den-bachelor/#comment-134>)

ARCHIV

Januar 2018 (<http://sebinside.de/2018/01/>)

Juni 2017 (<http://sebinside.de/2017/06/>)

Mai 2017 (<http://sebinside.de/2017/05/>)

April 2017 (<http://sebinside.de/2017/04/>)

Oktober 2016 (<http://sebinside.de/2016/10/>)

September 2016 (<http://sebinside.de/2016/09/>)

August 2016 (<http://sebinside.de/2016/08/>)

Juli 2016 (<http://sebinside.de/2016/07/>)

Juni 2016 (<http://sebinside.de/2016/06/>)


April 2016 (<http://sebinside.de/2016/04/>)

März 2016 (<http://sebinside.de/2016/03/>)

SOCIAL MEDIA

(<http://sebinside.de/2017/06/26/lets-play-automatisierung/>)
 (<http://facebook.com/sebinside>)
 (<http://youtube.com/sebinside>)
 (<http://github.com/sebinside>)

FEED

 (<http://sebinside.de/feed/>) RSS - Beiträge (<http://sebinside.de/feed/>)

 (<http://sebinside.de/comments/feed/>) RSS - Kommentare (<http://sebinside.de/comments/feed/>)



Ich heie Sebastian, studiere Informatik am KIT in Karlsruhe und betreibe den YouTube Kanal skate702 (<http://youtube.com/skate702>). Auf diesem Blog geht es um Technik, Software und Minecraft!

Impressum (<http://skate702.de/impressum/>) © 2016 skate702 – Theme von Colorlib (<http://colorlib.com/>) Powered by
WordPress (<http://wordpress.org/>)