|  |
| --- |
|  |
| SpeedClick |
| Der privateste Browser |

|  |
| --- |
| Bledion Morina  11.5.2023 |



# Inhaltsverzeichnis

[0. Inhaltsverzeichnis 1](#_Toc134693440)

[1. Vorwort 2](#_Toc134693441)

[2. Hauptteil 3](#_Toc134693442)

[2.1 Python und die Frameworks 3](#_Toc134693443)

[2.2 Design des Browsers 4](#_Toc134693444)

[2.3 Funktion des Browsers 5](#_Toc134693445)

[2.4 Die Website 6](#_Toc134693446)

[2.5 Die Hindernisse und Limitationen 7](#_Toc134693447)

[3. Zusammenfassung und Reflektion 8](#_Toc134693448)

[4. Quellverzeichnis 9](#_Toc134693449)

[5. Anhang 10](#_Toc134693450)

# Vorwort

Bevor ich die Idee hatte, den Browser zu erstellen, habe ich mir überlegt eine Videospielkonsole zu machen. Der Grund, wieso ich das nicht gemacht habe, war dass es viel zu Teuer war. Eine Konsole zu machen die mit den Heutigen Standards für Gaming entspricht würde knapp 1000.00 CHF Kosten. Dann habe ich mir eine Weile lang überlegt, was ich machen sollte, und bin zum Entschluss gekommen, dass ich etwas programmiere. Zurzeit wusste ich noch nicht was ich programmieren wird, aber es war sicher mal ein Anfang. Zuerst wollte ich ein Spiel programmieren aber habe gemerkt, dass dies nicht nur lange geht, sondern auch viel Arbeit erfordert. Ich habe dann drüber nach gedacht eine Wetterapp zu mache wo dir den Wetter anzeigt aber nicht nur ist, dass viel zu einfach, sondern auch ziemlich beliebt und somit würden, dass viel zu viele Leute machen. In diesem Moment ist mir eine geniale Idee aufgetaucht. Ich mache einen Browser. Die Überlegung dahinter war, dass es etwas wirklich Gebräuchliches ist und es auch wenige davon gibt. Es ist etwas eher Schwierigeres zu machen und daher schrecken sich Leute davon eher ab. Ich nicht, ich habe mir die Herausforderung angenommen einen Browser zu machen aber nicht nur irgendeinen, sondern den sichersten und privatesten Browser, den es gibt. Ich habe mir schon den Allgemeinen design schon vorgestellt und war bereit es anzufangen. Ich habe mir zuerst noch ein bisschen Vorwissen angesammelt die ich brauche, wie zum Beispiel den Framework PyQt5 gelernt, was der Hauptbaustein dieses Browsers sein wird. Es war eine Herausforderung da der andere Framework Tkinter und PyQt5 nicht so gut zusammenarbeiten. Ich musste dann halt alles auf PyQt5 aufbauen, was ich natürlich nicht kannte. Als ich aber angefangen habe es zu lernen musste ich feststellen, dass es relativ einfach war zu lernen. Ich konnte wichtige dinge wie die Labels und Webview lernen. Andere wichtige Frameworks waren auch: Sys, Os, Pyautogui und Keyboard. Sys wird gebraucht, um auf wichtige Systemressourcen zugreifen zu können, Os ist, um die Informationen des Operating Systems zu wissen, Pyautogui wird für Errorlogging gebraucht und Keyboard für die Tastenkombinationen und Input für den Browser. Als Programmiersprache habe ich Python gewählt da es relativ einfach ist und somit im Zeitrahmen einfach erledigt wird. Somit habe ich mir die Frage gestellt ist es möglich einen Browser in Python zu erstellen.

# Hauptteil

## Python und die Frameworks

Als ich angefangen habe musste ich mir zuerst überlegen in welcher Programmiersprache ich meinen Browser programmieren werde. Da habe ich mir viele Überlegungen machen müssen und zuerst war ich mit voller Meinung überzeugt, dass ich es in C# oder Java machen werde. Dann musste ich leider feststellen, dass es schwierige Frameworks sind und ich es mit der Gegebenen Zeit vielleicht nicht schaffen werde. Dann ist es mir im Sinn gekommen, Python. Es ist eine einfache Sprache die mehr für KI, Automationen und Bots verwendet wird. Natürlich aber gab es den Framework PyQt5 was mir einen Webview gegeben hat, eine Applikation und Labels, also genau, dass was ich brauche. Ab diesem Moment musste ich vieles lernen wie zum Beispiel die Labels. Labels sind Knöpfe, Texte und weiteres die in einer App für die Funktionalität dient. Man kann bei einer App ganz einfach einen Knopf rein machen und dann mit der Funktion, button.clicked.connect(Funktion), die knöpfe mit Python Funktions verbinden. Python Funktions sind eine Reihe von Code Schnipsel, die in eine Funktion reingepackt werden, sodass man sie jederzeit mit Funktion() wieder aufrufen kann. Somit kann man sich unendlich viel Arbeit sparen. Die Webview ist da, um eine Website anzuzeigen. Eine Website kann angezeigt werden, wenn man mit QUrl(WebsiteUrl) eine Website mit der Webview verbindet. Dabei ist wichtig zu merken, dass QUrl links keine Strings sind. Bei allen Programmiersprachen gibt es Variablen, wie in der Mathematik. Speziell bei den Variablen der Programmiersprachen ist, dass man ihre datentype angeben muss, was so viel heisst, wie wenn man einen Text speichern will, muss man angeben, dass es ein Text ist. Die verschiedenen Typen sind folgende: String = Text, Integer = Zahl, Float = Dezimalzahl und Boolean = wahr/falsch. Wichtig dabei ist, dass die verschiedenen Typen ihre eigenen Limitationen haben. Wenn man jetzt einen Text haben will, muss man angeben str Variable = „Text“. Bei Python ist der Vorteil, dass es Automatisch erkannt wird was für ein Typ es ist das heisst man kann einfach eingeben Variable = „Text“. Um die Frameworks zu importieren habe ich ganz oben im Code einfach eingeben müssen, Import PyQt5. Auch hier kann man die Imports verschieden haben wie z.b man will nur die Labels von PyQt5, from PyQt5 Import Labels. Es gab ein paar Probleme, die mir aufgetaucht sind wie zum Beispiel, dass ich die Webview eine Weile lang nicht Vergrössern oder verkleinern lies oder auch mit den Tabs da ich nicht wirklich wollte, dass sie viel RAM verbrauchen, aber auch dass sie Speichern und man ohne Probleme Tabs ändern kann, ohne dass irgendwie alles, was man gemacht hat, verloren ging. Ich konnte diesem Problem hintergehen, in dem ich einen Savestate des Tabs gemacht habe. Damit ist gemeint, dass ich den Jetzigen Stand des Tabs komplett gespeichert habe und es jederzeit wieder Aufrufen konnte.

## Design des Browsers

Das Erste, was man in einem Browser oder auch allgemein App sieht, ist das Design. Es beschreibt die Art des Browsers und nicht nur muss es Einfach zu bedienen sein, sondern auch Herausstechen. Dann nach einigem überlegem bin ich mit einem bisschen älterem design gegangen da dies vor allem herausstecht. Die neuen Designs sehen all ein bisschen zu simple aus. Dort bin ich ein bisschen Internet Explorer Style gegangen aber nicht all zu viel da Internet Explorer zu viele Knöpfe hatte und dies einfach nur unnötig und ablenkend ist. Schlussendlich bin ich mit meinem simplen Design zufrieden. Was aber wichtiger als das Design ist das Layout der verschiedenen Knöpfe und Tabs. Die Tabs könnte man wie bei Opera Links machen oder auch oben wie in den meisten Browser. Ich bin schlussendlich mit den Tabs oben mehr zufrieden da es klassisch ist und es in praktisch jedem Browser so vorhanden ist. Die Art und Weise wie man neue Tabs öffnet ist in dem man in einen Leeren Feld, wo normalerweise die Tabs sind, einfach doppelklickt oder einfach auf das + bei Dateien drückt. Bei meinem Browser sollen Anstatt das ganz oben die Tabs kommen einfach die Suchleiste und Weiter Knöpfe zuerst kommen und danach die Tabs was für ein Einfacheres Layout dient. Was noch wichtiger ist sind die Icons. Die musste ich auch Designen und habe mich auf ein Blockiges aber auch Klassisches design entschieden. So sind schlussendlich diese Knöpfe entstanden. Ich konnte zum Glück mein vorheriges Wissen zu Pixel Art zugute machen die ich früher für Videospiele brauchte. Ich kannte somit viele Techniken wie zum Beispiel das Shadowing. Bei Erstellen der Icons ist mir aufgefallen, dass sie von der Grösse her mit den Knöpfen passen müssen, da sonst ein Anteil herausgeschnitten wird. Somit bin ich mit der Grösse von 50x50 Pixel gegangen. Klingt jetzt nach viel in der Mathematik aber ein Standardbildschirm beträgt eine Grösse von 1980x1080 Pixel. Der Vorteil bei so kleinen Icons ist, dass die Ecken mit ein bisschen Shadowing rund aussehen, obwohl es eckig ist. Ich habe die Icons dann auf eine Grösse von 25x25 runtergestellt was einen runderen Effekt abgeben wird. Für die Tabs habe ich zuerst auch an Icons gedacht, bin aber zum Entschluss gekommen, dass der Titel der Website besser hineinpasst. Möglicherweise werde ich in einem Zukünftigen Update noch das Icon des Besuchten Websites hinein machen. Bei den Tabs ist mir noch eingefallen, dass ich ihnen ein Design geben muss, weil sie sonst eckig aussehen. Ich habe mit überlegt etwas ähnliches wie Firefox zu machen da ich dieses Design schön finde. Ich bin schlussendlich zufrieden wie es mit meinem Design gekommen bin. Es ist herausstechend und auch Simple gehalten. Etwas Interessantes, was ich eingebaut habe, ist die Funktion die Suchleiste und die Knöpfe so zu verschieben, wie man will. Man kann sie links, rechts oder oben zu behalten. Es besteht auch die Möglichkeit sie als ein eigenes Window zu haben. Das habe ich mit QActions geschafft.

## Funktion des Browsers

Das absolut wichtigste einer Applikation, in meinem Beispiel mein Browser, ist die Funktion. Der Browser bring absolut nichts, wenn man nicht die verschiedenen Knöpfe drücken kann und sie dann eine Funktion aufrufen. Die Funktionen sind der Grösste Bestandteil meines Browsers. Somit sind sie auch das, womit ich am meisten Kämpfen musste. Ich musste verschiedene Herausforderungen widmen und sie dann Lösen. Was mir am meisten Probleme bereitet hat, waren die Tabs. Wie schon erwähnt, sind Tabs grosse Ressourcen Verbraucher. Das Problem ist, dass eine offene Applikation im RAM (Random Access Memory) gespeichert wird. Ein normaler Rechner hat 4-8gb, mittlerweile schon 16gb. Da jeder einzelne Tab einen Webview offen hat, also eine Website, wird dies im RAM gespeichert. Um das zu umkämpfen, musste ich nach mehreren Lösungen im Internet suchen und nach langem überlegen bin ich zum Schluss gekommen, wie schon erwähnt, die Tabs in einem Savestate zu speichern. Der Vorteil dabei ist, dass anstatt bei 3 Tabs 3 Websites offen sind, ist nur eine Website geöffnet und bei den anderen wird nur der Jetzige Zustand gespeichert. Jetzt kommt sicher die Frage vor, wieso ist denn der RAM so wichtig? Für neue Rechner wird der RAM sicherlich kein Problem sein aber da es immer noch eher ältere Rechner gibt, muss ich schauen, dass ich die 2gb RAM nicht überschreite, da sonst das ganze System abstürzen würde. So weit haben wir nur über die Tabs geredet. Weitere wichtige Funktionen sind die Knöpfe. Ich finde die Knöpfe sind wichtiger als die Tabs. Bei den Knöpfen hatte ich zum Glück nicht so viele Probleme da sie einfach waren. Um eine Funktion einzubinden, musste ich nur button.clicked.connect(Funktion()) eingeben. Um die Funktion einzubinden, wird auch eine Funktion gebraucht, die aber von PyQt5 gegeben ist. Das komplizierte aufzusetzen war die Webview, weil sie mir viele Probleme vorbereitet hat. Die Tabs waren schwieriger aber auch nur weil die Webview nicht wirklich mitgemacht hat. Die Webview aufzustellen war einfach aber nachher die verschiedenen Funktionen damit zu binden war schwierig. Aus der Seite von PyQt5 wird nicht wirklich viel dafür angeboten und man muss fast schon eigene Codes erfinden. Ich konnte zum Beispiel die Webview nicht vergrössern oder verkleinern da die PyQt5 Centralwidget es mit nicht erlaubt hat. Centralwidget ist sehr wichtig da dies der Baustein für alle Widgets und Labels sind. Wichtig dabei ist, dass ich nicht Widgets benutzt habe. Normalerweise werden Widgets gebraucht, aber Labels lagen in meiner Situation besser da sie einfacher sind mit Funktionen zu verbinden. Labels sind einfach Texte, die man ändern kann, wie man es braucht. Ich konnte schnell feststellen, dass der Webview wirklich langsam ist und somit musste ich im Internet nach Lösungen suchen. Dort bin ich auf einen Post auf Stack Overflow gekommen und konnte den code nehmen. Leider musste ich den Ganzen „Kopierten“ Code komplett umschreiben da es mit meinem Code nicht zusammen passte.

## Die Website

Mein Browser hat im Moment leider noch keinen eigenen Webcrawler, sondern benutzt DuckDuckGo als Suchmaschine. Mein Browser ist schlussendlich nur da, um eine Website anzuzeigen. Ich habe schon verschiedene Lösungen versucht und bin sogar so weit gegangen, dass ich einen ganzen Code genommen habe und in meine Website gepackt habe, damit es Websites suchen kann. Leider musste ich feststellen, dass es von Google basiert und somit den Titel, der Privateste Browser, verlieren würde. So bin ich zum Entschluss gekommen, DuckDuckGo zu benutzen, da es praktisch, wie Google funktioniert aber keine daten von den Benutzern verfolgt. Wenn man bei meiner suchleiste jetzt zum Beispiel Youtube eingibt, dann würde es automatisch auf DuckDuckGo suchen. Etwas Wichtiges, was ich auch gemacht habe, war, dass man bei der Suchleiste nicht zum Beispiel <https://youtube.com/> eingeben muss, sondern direkt „youtube.com“ oder youtube eingeben kann. Leider musste ich die Webcrawl Idee aus Privatsphäre gründen eliminieren. Um die Website zu designen habe ich sites.google.com benutzt, um die Website zu erstellen, weil ich nicht so viel Zeit hatte, um noch eine Perfekt Website zu erstellen. Ich habe eingebaut, dass wenn man mein Browser startet, dass es direkt als erstes meine Website öffnet. Wenn man aber neue Tabs öffnet dann eben nicht. In einer Zukünftigen Version werde ich dies noch einbauen, nur der unterschied, dass es nicht von Google basiert aber von DuckDuckGo. Das wäre aber fascht schon eine eigene Projektarbeit. Das tolle dabei ist, dass ich nicht wirklich um Erlaubnis fragen muss, da die meisten Webcrawler auf der GNU-Lizenz sind und somit gratis zum Gebrauchen und Kopieren. Im Moment liegt mein Browser auch noch in der GNU-Lizensierung und ich werde es auch nie ändern, da ich möchte, dass Leute meinen Browser Kopieren können und so bearbeiten wie sie wollen. Da DuckDuckGo im Moment noch closed Source ist, kann ich dies leider noch nicht vollenden. Das grossartige ist, dass sie im Moment immer mehr auf Open Source wechseln und somit vielen anderen Entwicklern wir mir Chance bereiten, so etwas zu erstellen, was für meinem Browser Vorteilhaft wäre. Meine grösstes Problem dabei ist, dass ich dann mit JavaScript weitermachen müsste, damit die Website funktioniert. Ansonsten ist die Website modern und Simple geworden. Ich habe noch nicht viel geschrieben dort, aber ich habe mal den Download link hinein gemacht was den Benutzern zu meiner GitHub Page führt. Dort kann er dann alle Versionen sich ansehen, herunterladen oder bearbeiten. Das Schöne an GitHub ist, dass jeder Benutzer mein Projekt herunterladen kann, bearbeiten kann wie er will und dann wieder Hochladen und einen Pushrequest ziehen. Wenn es mit gefallen würde, was der Benutzer gemacht hat, kann ich seine Version mit meiner Version verbinden und so entsteht ein neues Update mit individuellen Funktionen anderer Benutzer.

## Die Hindernisse und Limitationen

Eines der grössten Herausforderungen im Programmieren sind die Limitationen. In meinem Beispiel, wie erwähnt, konnte man klar merken, dass ich einige Hindernisse wie die RAM-Nutzung hatte. Das grösste, was einem Entwickler durch den Kopf geht, ist, wie gut kann meine Applikation auf einem Schwächeren Rechner laufen? Bei den meisten Browser ist es so, dass die Entwickler mit dem Kämpfen müssen. Da mein Browser aber so leicht ist, kann es ziemlich gut auch schwächeren Rechner laufen. Um die RAM-Auslastung auszugehen habe ich im zu vergleich Google, statt die Tabs die ganze zeit offen zu haben, speichere ich ihren jetzigen stand ab und hole es wieder, wenn ich es brauche. Was ich dabei merken musste, ist, dass ich dann keine Videos mehr im Hintergrund laufen lassen kann. Um das zu umgehen habe ich mit dem Videoplayback code von PyQt5, erkennen können, ob ein Video im Hintergrund läuft und es dann in einer Separaten App offen gehabt, die im versteckt im Hintergrund liegt. Somit war es möglich Videos in anderen Tabs offen zu haben und sie weiterhin hören zu können. Das Gleiche gilt auch für Audios. Ein weiteres Problem, was ich bis jetzt noch nicht angesprochen habe, sind die Funktionen Websites als HTML zu speichern und sie wieder öffnen. Das mit dem Öffnen ist etwas komischer, weil es einmal geht und dann auf einmal nicht mehr. Websites zu drucken war auch eine Funktion, die ich angestrebt habe, jedoch musste ich feststellen das die Browser klasse mit dem Windows Konflikte hat. Diese Probleme konnte ich mit reue leider nicht mehr erledigen, da sie so dermassen gross sind, dass ich wahrscheinlich noch Wochen gebraucht hätte, um auf eine Lösung zu kommen. Allerdings konnte ich die Tabs noch retten. Wie ich schon ein paar Mal erwähnt habe, waren Tabs ein wirkliches Abenteuer. Zuerst war beim Problem, wie kann ich Tabs erstellen und löschen? Da habe ich einen Post auf Stack Overflow gesehen, was QBoxLayouts anspricht. Ich konnte die Idee mit mir nehmen und habe einen QBoxLayout erstellt und habe direkt einen Knopf gemacht, der andere Knöpfe erstellt. Im gleichen Post kamen noch die Funktionen in Aussage. Dort habe ich gesehen, dass mit einer Lambda Funktion, nicht mal dynamische Funktionen erstellen musste, sondern einfach für jeden Knopf eine eigene Funktion hatte. In der Funktion war gegeben, dass der Computer in einer Erstellten Liste, die Position des Tabs in der Liste findet und die Website dann nach der Liste ladet. In der Liste waren die Websites in der Reihenfolge der Tabs zugeordnet, das heisst, wenn der Tab die Position 1 hatte, war in der Liste, bei der Position 1, die Website des ersten Tabs gespeichert. So habe ich den Prototyp meiner Tabs erstellen können. Nach einer weile habe ich eine effektivere Lösung gefunden, die nicht von einer Liste basiert.

# Zusammenfassung und Reflektion

Insgesamt ist meine Projektarbeit gut geworden. Ich hätte vielleicht ein paar Sachen mehr machen können wie zum Beispiel die Funktion etwas herunterladen zu können aber so weit bin ich zufrieden. Ich hätte vielleicht etwas mehr an meiner Projektarbeit arbeiten müssen damit es Perfekt geworden wäre, leider hatte ich nicht genug Zeit, um dies zu tun. Ich finde, dass ich mit mehr Zeit auch mehr hinbekommen hätte, wobei ich vielleicht auch einfach einen besseren Zeitplan hätte erstellen können. Leider konnte ich keine VPN mehr einbauen, was ziemlich gut gewesen währe für die Privatsphäre. Bis jetzt hat mein Browser Tabs, eine Suchleiste, Refresh/Return/Forward Knopf, einen cancel loading Knopf, was eine Verbindung zu einer Website unterbricht, einen Home Knopf der den Benutzer zu meiner Homepage bring, die Funktion auf der suchleiste Wörter zu suchen auf DuckDuckGo, ohne dass es probiert eine Website zu öffnen, eine Funktion HTML code zu laden, was allerdings noch in Bearbeitung ist und Websites zu speichern. Das sind viele Funktionen, die auch meine Leitfrage beantworten. Somit wurde die Leitfrage, kann man einen Modernen Browser in Python machen, beantwortet. Die Antwort ist ja, mit viel Zeit und Mühe. Programmieren ist ein schwieriges Hobby aber die Belohnung, die man dafür kriegt, ist sehr gross. Man konnte bei meiner Projektarbeit schon sehen, ich habe meinen eigenen Browser gemacht, ich weiss alles, was im Hintergrund passiert und weiss, dass ich nicht in irgendeiner Art und Weise ab verfolgt wird. Wobei man sagen muss, wenn man sich mit dem Internet verbindet, dann wissen natürlich die Website Anbieter deine Informationen und nicht der Browser. Ein Browser ist nur da, um sich mit dem Internet zu verbinden und ist nicht zu verwechseln mit einer Website. Meine Website finde ich ist auch gut rausgekommen wobei ich noch versuchen werde eine Search Engine reinzubauen die auf DuckDuckGo basiert. Vielleicht werde ich das schon kurz nach der Projektarbeit einbauen. Ich finde das großartige an meiner Projektarbeit ist, dass ich jederzeit weiter daran arbeiten kann und meine Bedürfnisse hinein machen. Ich werde auch die Zukünftigen Versionen auf GitHub weiterhin veröffentlichen. Meine Website will ich dann auch noch erweitern. Etwas, was ich aber an der Website ändern will, ist das Design. Es passt nicht so gut mit dem „alten“ style meines Browsers. Die RAM-Nutzung mir den Tabs konnte ich auch gut umgehen vor allem mit der Methode von Save States. Somit kann man die RAM-Nutzung um etwa 80% eliminieren. Dass ich die Grösse der Webview verstellen konnte, war eines der wichtigsten Funktionen für mich. Ich finde, dass meine Projektarbeit schwierig war zum Erstellen, aber schlussendlich hat es sich sehr gelohnt. Mit meinem Browser kann ich jetzt mit weniger Gedanken im Hinterkopf ins Internet surfen, ohne mir wirklich viel sorgen über meine Privatsphäre zu machen.

# Quellverzeichnis

* Stack Overflow, Clicking a button to create a new button using PyQt5 python, <https://stackoverflow.com/questions/58471579/clicking-a-button-to-create-a-new-button-using-pyqt5-python>, 24.4.2023.
* PyQt5, Qt for Python, <https://doc.qt.io/qtforpython-5/>, 20.4.2023.
* Stack Overflow, Pandas: Count dots in a string- same length?, <https://stackoverflow.com/questions/59704959/pandas-count-dots-in-a-string-same-as-length>, 1.5.2023.
* Naishare, How to build a simple search engine using HTML, CSS and JavaScript?, <https://www.naishare.com/blog/how-to-build-a-simple-search-engine-using-html-css-and-javascript>, 17.4.2023.
* W3Schools, how to use Python Classes, <https://www.w3schools.com/python/python_classes.asp>, 20.2.2023.

# Anhang

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Software, Zahl enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung