

Realisierung des Spieleklassikers "Archon"

mit 3D– und Webtechnologien

Bachelorarbeit

im Fachgebiet Software-Engineering

zur Erlangung des akademischen Grades

Bachelor in Engineering



Vorgelegt von:	Kevin Dyes
Matrikelnummer:	2694420
Hochschule:	Technische Hochschule Georg-Simon-Ohm
Studienbereich:	Elektro– und Informationstechnik
Erstgutachter:	Prof. Dr. Röttger
Zweitgutachter:	Prof. Dr. Hopf

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Vorwort	III
Erklärung	V
1 Einleitung	1
1.1 Hinführung	1
1.2 Motivation / Problemstellung	1
1.3 Ziele dieser Arbeit	2
1.4 Aktueller Forschungsstand	2
1.5 Zentrale Begriffe	2
1.5.1 Was versteht man unter Webtechnologien?	2
1.5.2 Was versteht man unter 3D-Technologien?	3
1.6 Aufbau	3
2 Hauptteil	5
2.1 Einleitung	5
2.2 Analyse	5
2.2.1 Analyse des klassischen Spiels	5
2.2.2 Anforderungsanalyse an neue Implementierung	7
2.3 Implementierung	7
2.3.1 Architektur	7
2.3.2 benötigte Technologien und Frameworks	8
2.3.3 Hilfsmittel und Vereinfachungen	8
2.3.4 Schritte der Implementierung	8
2.4 Resultate	8
2.4.1 fertige Architektur	8
2.4.2 Erfüllung der Anforderungen	8
2.4.3 Überprüfung der Software mit Unit-Tests	9
2.5 Fazit	9
3 Fazit und Ausblick	11
3.1 Fazit	11
3.1.1 erreichte Ziele	11
3.2 Ausblick	11

Verzeichnisse	VII
Literaturverzeichnis	VII
Abbildungsverzeichnis	IX
Tabellenverzeichnis	XI
Quellcodeverzeichnis	XIII
Abkürzungsverzeichnis	XV
Stichwortverzeichnis	XIX

Vorwort

Im Jahr 2017 wurden mehr als 1,8 Milliarden browser-fähige Endgeräte weltweit verkauft. Der technologische Fortschritt in Hardware und Software erlauben es dieser großen Masse an Zugängen zum Internet eine große Vielfalt an Applikationen nahezu überall zu nutzen. Seien es Business-, oder Unterhaltungs-Applikationen, und auch Spiele jeglicher Art.

Die klaren Vorteile des Web als Plattform, auch für Spiele, sind die Offenheit, der einfache Zugang und die immer weiter steigende Unterstützung neuer Technologien.

Sogenannte Spiele-Engines, also Bibliotheken die sich um die meisten, generellen Problemstellungen eines Spiels und seiner Entwicklung kümmern, gibt es daher allerlei. So sind heutzutage hardwarebeschleunigte 3D-Visualisierung und Echtzeitkommunikation im Webbrowser Stand der Technik. In dieser Arbeit soll die Verschmelzung von State of the Art Webtechnologien mit einem Retro-Spiel erfolgen.

"Archon" genoss für Atari und C64 großen Erfolg, ähnlich vieler anderer Spiele für die Konsolen der 80er. Durch die strategischen und actionreichen Aspekte in einem Spiel hat es viel Abwechslung, auch aus Sicht eines potentiellen Entwicklers für einen Ableger des Spiels. Das Spiel dient daher als sehr gutes Exempel für die Möglichkeiten des Webs als Spieleplattform, aber auch der Möglichkeiten und Freiheiten des Webs für jede andere Art von Applikation.

Anmerkung:
Recherche
Erfolg von
Archon belegt

Erklärung

Ich, Kevin Dyes / Matrikel Nummer 2694420, versichere hiermit, dass ich diese Bachelorarbeit mit dem Thema

Realisierung des Spieleklassikers "Archon"
mit 3D- und Webtechnologien

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Nürnberg, den 25. Mai 2018

Kevin Dyes

1 Einleitung

1.1 Hinführung

Die Hinführung soll den Leser an das Thema 3D-Technologien, Web-Entwicklung und Spieleklassiker, wie Archon heranführen. Hier sind Quellen und Recherchen zu ein paar Zahlen, wie Remakes, Umsätze zu Uralt-Spielen aktuell und Beliebtheit/Zahlen zum Webbrowser als Spieleplattform angebracht, um Wichtigkeit der Plattform und Technologien, aber auch Relevanz für Spieleklassiker zu zeigen.

- Verkaufte Konsolen
- Produzierte Spiele und Verkäufe
- Neuauflagen von Spielen und Konsolen
- Neuauflagen von Archon
- Virtualisierung und Emulatoren um alte Spiele zum Anfassen zu bekommen
- Aber, nur auf richtigen PCs zu finden und meist nicht wirklich legal
- Web als freie, riesige Plattform mit großem Zugriffsbereich
- Web APIs, die in den letzten Jahren erschienen sind (kurze Recherche zu Arbeiten des W3C!)
- Schiere Vielfalt an JS-Frameworks zu Web APIs und Komfort bei der Entwicklung (npm packages um Zahlen zu bekommen)
- Beispiele zu 3D-Möglichkeiten im Web, Stichwort: THREE.JS-Demos

1.2 Motivation / Problemstellung

Diese Sektion soll vom vorherigen Stand der Forschung und Möglichkeiten auf Anwendbarkeit überleiten und zeigen, dass die evtl. surreal erscheinende Vermischung von Genres, Plattformen und Technologien anwendbar ist, und das sogar mit beschränktem Aufwand. Vereinbarkeit von aktuellen Technologien mit Spiele-Klassikern und Spiele-Klassiker als Anwendungsbeispiele dafür. Bisher sind oftmals nur Demonstrationen der einzelnen Technologien zu finden, oder kurze Starter-Kits bzw. Einstiegspunkte in die jeweilige Technologie, ...

Versionen, Ableger und Kopien des Spieleklassikers Archon gibt es zahlreich. Meist sind diese jedoch auf ein System beschränkt, oder tatsächlich nur mit einem Emulator ausführbar. Das Web leistet an dieser Stelle die perfekte Abhilfe: Es gibt unzählige Geräte, die heutzutage einen

Webbrowser installiert haben, somit hat jeder Zugriff auf diesen Ableger, der ein webfähiges Gerät besitzt.

1.3 Ziele dieser Arbeit

Hier soll bloß kurz und prägnant die Fragestellung genannt werden und eine kurze Analyse der einzelnen Worte des Titels stattfinden. Ein Satz, wie "Die exakten Anforderungen für die Umsetzung dieser groben Ziele werden zu einem späteren Zeitpunkt herausgearbeitet", oder "... Anforderungen werden aus dem Aufbau der Arbeit heraus klar." muss hier hinein.

1.4 Aktueller Forschungsstand

Hier sollen dann Extrema zum einen in Richtung 3D-Entwicklung, in Richtung Web-Entwicklung und aber auch bei der Spiele-Entwicklung kurz herausgearbeitet werden, um dem Leser einen kurzen Einblick in aktuelle Möglichkeiten zu geben, und zu zeigen, dass die Arbeit "State of the Art" ist.

Fliegt vermutlich raus, oder kommt an andere Position.

1.5 Zentrale Begriffe

1.5.1 Was versteht man unter Webtechnologien?

Webtechnologien betiteln die Sammlung aller nötigen Aspekte zum Erstellen einer Webanwendung. Webanwendungen selbst bestehen aus einem Client-Server-Modell. Typische Bestandteile des Clients sind:

- HTML zur Beschreibung des Inhalts
- CSS zur Beschreibung des Aussehens
- JavaScript zur Dynamisierung des Clients

Ein Webserver gibt auf HTTP-Anfragen die entsprechenden Inhalte und Medien an den Client heraus, welcher durch einen Webbrowser angezeigt wird. Außerdem können hier Daten des Clients verarbeitet, gespeichert und verteilt werden werden.

1.5.2 Was versteht man unter 3D-Technologien?

Da ein Bildschirm, beispielweise eines Computers, nur zweidimensional ist, muss durch andere Methodiken der Effekt einer dritten Dimension geschaffen werden.

Objekte im 3D-Raum werden über ihre Eckpunkte aufgespannt und im Code somit in einem 3D-Raum mit normalem 3-Achsen-Koordinatensystem dargestellt. Anschließend folgt der Prozess des Renderns, bei dem zunächst aus den einzelnen Punkten der Objekte die Formen errechnet werden, indem die Eckpunkte zu Flächen verbunden werden. Anschließend erfolgt eine Ausrichtung aller Flächenpunkte am Pixel-Raster, sodass eine 3D-Projektion der 2D Pixel entsteht. Der nächste Schritt, Fragment-Bearbeitung, behandelt die Einfärbung der, im vorherigen Schritt gebildeten, "Fragmente" anhand von Licht und verwendeten Texturen. Der finale Schritt des Renderns wandelt die 3D-Projektion in ein 2D-Pixel-Bild, dass dann auf dem Bildschirm angezeigt wird. Außerdem werden hier Prüfungen für die Sichtbarkeit von Objekten unternommen, sodass nicht sichtbare Objekte, oder Teile von ihnen, auch nicht weiter verarbeitet werden.

1.6 Aufbau

Hier wird dann auf den Aufbau des Haupt- und Schlussteils eingegangen. Also der erste Abschnitt beschäftigt sich mit Anforderungsanalyse beider Plattformen/Seiten. Der zweite Abschnitt dreht sich dann rund um Zusammenführung der Anforderungen und Tool-Auswahl, und somit darum einen Einstiegspunkt und eine Architektur für die weitere Entwicklung festzulegen.

Im ersten Teil dieser Arbeit sollen die Anforderungen an eine Neuauflage von "Archon" erfasst werden und eine Beschreibung der Features und Spielmechaniken zur Umsetzung erstellt werden. Anschließend werden die Anforderungen der technologischen Seite herausgearbeitet, sodass eine Liste an Funktionalitäten entsteht, die von den 3D- und Webtechnologien erfüllt werden muss. Alle Anforderungen werden dann in eine Beschreibung zur Umsetzung und eine entsprechende Architektur ausgearbeitet. Den Abschluss bildet eine Überprüfung der fertigen Neuauflage auf die Erfüllung der Anforderungen, sowie eine Präsentation und eine Reflexion des Ergebnisses.

2 Hauptteil

2.1 Einleitung

2.2 Analyse

The goal of every video game is to present the user(s) with a situation, accept their input, interpret those signals into actions, and calculate a new situation resulting from those acts. Games are constantly looping through these stages, over and over, until some end condition occurs (such as winning, losing, or exiting to go to bed). Not surprisingly, this pattern corresponds to how a game engine is programmed. The specifics depend on the game. [13]

In diesem Abschnitt soll die Vorgehensweise bei der Analyse kurz aufgezeigt werden und der Inhalt der einzelnen Teile klar gemacht werden. Also, zum einen, dass hier eben eine Anforderungsanalyse stattfindet, und zum Anderen, dass diese Analyse zweigeteilt ist.

2.2.1 Analyse des klassischen Spiels

Hier werden alle Aspekte des klassischen Spiels herausgearbeitet, gegliedert nach Teilen: Generelle Regeln, das Board, die Figuren, Abläufe und Wirkungen, mögliche Einstellungen
Stichwort: Komponentendiagramm!

In diesem Abschnitt sollen nun die Anforderungen und Regeln aus Archon herausgearbeitet werden. Die Anforderungen und Regeln werden, der Übersicht halber, in folgende Kategorien unterteilt:

- Generelle Regeln und Ziele des Spiels
- Spieleinstellungen und Start
- Das Spielbrett
- Das Kampfareal
- Die Figuren

Die folgende Abschnitte enthalten eine kurze Zusammenfassung des Spiels und seinen Inhalten und Regeln anhand der obigen Aufteilung. Anschließend wird kurz zusammengefasst, welche Eigenschaften eine Realisierung des Spiels inne haben muss, um das Spiel ausreichend widerzuspiegeln.

Generelle Regeln und Ziele des Spiels

Archon ist ein Spiel, dass auf einem 9 x 9 Schachbrett stattfindet. Ähnlich wie beim Schach gibt es zwei Parteien, Licht und Dunkelheit, die sich im Wettstreit gegenüber stehen. Die Spieler ziehen dabei immer abwechselnd ihre Figuren auf dem Spielbrett, wobei kein Zug gepasst werden kann. Jeder Spieler beginnt mit 18 Figuren, die in acht verschiedene Typen gegliedert sind. Die Figuren von Licht und Dunkelheit sind vollständig unterschiedlich. Treffen zwei Figuren auf dem Spielbrett zusammen, stehen sie sich in einem Kampfareal gegenüber. Jede Figur hat dabei ihre eigenen Lebenspunkte und ihre eigene Angriffsstärke. Der Sieger dieses Kampfes bleibt auf dem Spielbrett, während der Verlierer aus dem Spiel genommen wird. Sollte der Kampf in einem Unentschieden ausgehen, werden beide Figuren aus dem Spiel genommen.

Das generelle Ziel des Spiels ist es alle 5 speziellen Machtfelder gleichzeitig mit eigenen Figuren zu besetzen. Desweiteren kann das Spiel gewonnen werden, indem alle gegenerischen Figuren besiegt werden, oder die letzte Figur mit einem Gefängnis-Zauber belegt wird. Ein Unentschieden tritt auf wenn das Spiel zu lange "passiv" ist, also wenn für mindestens zwölf Züge kein Kampf stattfindet und kein Zauber gewirkt wird.

Das Spielbrett

Das Spielbrett besteht aus 9x9 Feldern von drei Typen:

- Permanent weiße Felder
- Permanent schwarze Felder
- Felder, die zwischen Schwarz, vier Grün-Tönen und Weiß wechseln

Jeder Spieler hat einen Zug mit jeder Farbe, sodass die Farbe nach jedem Zug des zweiten Spielers wechselt. Ein Farbzyklus dauert damit zwölf Züge pro Seite: Sechs von Weiß zu Schwarz und Sechs zurück. Die Farben werden fortan mit Zahlen von 1 bis 6 durchnummeriert, wobei 1 Weiß darstellt und 6 Schwarz. Wenn die Licht-Seite das Spiel beginnt, startet der Farbzyklus bei Farbe 4 und wird dunkler. Beginnt die Dunkelheit-Seite, so startet der Zyklus bei Farbe 3 und wird heller.

Weiterhin gibt es die fünf Powerfelder, deren Einnahme die Siegbedingung darstellt. Diese Felder haben außerdem zwei spezielle Effekte auf Figuren, die darauf stehen. Zum einen können darauf stehende Figuren nicht Ziel eines Zaubers werden und zum Anderen werden die Figuren nach jedem eigenen Zug um einen gewissen Betrag geheilt. Diese fünf Powerfelder verteilen sich so, dass jeweils eines auf einem permanent schwarzen bzw. weißen Feld ist - dort wo der Zauberer bzw. die Zauberin ihre Ausgangsposition haben. Die anderen drei Felder sind auf farbwechselnden Feldern in der Mitte des Spielbretts verteilt.

Das Kampfareal

Das Kampfareal erscheint wenn zwei Figuren auf dem Spielbrett aufeinander treffen. Es stellt eine große Fläche dar, bei der von Zeit zu Zeit pflanzenähnliche Hindernisse erscheinen und wieder verschwinden. An den Seiten des Kampfareals werden die Lebenspunkte der kämpfenden Figuren in Form von Balken dargestellt.

Die Figuren können sich hier frei bewegen und haben, je nach Typ, unterschiedliche Eigenschaften, Attacken und Geschwindigkeiten.

Die Figuren

Die Zauber

2.2.2 Anforderungsanalyse an neue Implementierung

Hier wird analysiert, welche Aspekte das Spiel von der Seite der technologisch gestellten Anforderungen erfüllen muss. Also: Es muss eine 3D- Engine geben, die irgendwie im Browser renderbar ist.

Es muss möglich sein, alle Spiel-Mechaniken mittels Server-Client-Kommunikation umzusetzen, etc.

Stichwort: Komponentendiagramm!

2.3 Implementierung

Hier soll dann aufgezeigt werden, welche Anforderungen vom Spiel durch welche Technologie umgesetzt werden und nötige Bedingungen/grundsätzliche Ausschlüsse aufgezeigt werden.

Hier, oder im nächsten Abschnitt muss klar gestellt werden, dass ein KI-Modus nicht implementiert wird!

2.3.1 Architektur

Aus dem Teil der Zusammenführung ergibt sich dann zwangsläufig eine Art Komponentenarchitektur, die hier verfeinert wird, bis zu dem Punkt, an dem eine Festlegung auf Frameworks und Technologien getätigt werden muss.

2.3.2 benötigte Technologien und Frameworks

Hier wird dann von den benötigten Technologien die Festlegung auf eine bestimmte Implementierung getroffen, also Frameworks, Programmiersprachen, Toolsets etc. festgelegt.

Hier wird dann die Komponentenarchitektur vom vorherigen Teil weiter verfeinert, sodass evtl schon eine grobe Klassenarchitektur dabei herauspringen kann.

2.3.3 Hilfsmittel und Vereinfachungen

Hier würden Dinge zu lesen sein, wie die Benutzung von TypeScript, oder Browser-Beschränkungen etc, da diese nicht relevant für den eigentlichen Entwicklungsvorgang sind, aber dennoch nützlich sind und eben Vereinfachungen darstellen.

2.3.4 Schritte der Implementierung

Hier wird dann der Entwicklungsprozess kurz erläutert, also Vorgehen, wie Aufsetzen der Tool-Chain, dann Entwicklung einer ersten Darstellung (Frontend), um direkte Erfolge zu sehen, etc pp.

2.4 Resultate

Hier soll dann ein Screenshot des Ergebnisses rein und erläutert werden, dass als nächste der Endstand mit seiner Architektur gezeigt wird und anschließend die Erfüllung aller Anforderungen sichergestellt wird. Als letztes (falls genug Zeit!) werden die (hoffentlich) programmierten Unit-Tests erwähnt, und deren Ergebnisse dargestellt.

2.4.1 fertige Architektur

Die fertige Architektur kann und soll durchaus von der geplanten Abweichen und das schlussendliche Ergebnisse wird hier in Form von Diagrammen gezeigt, die dann einzeln erklärt werden.

2.4.2 Erfüllung der Anforderungen

Hier werden die Anforderungen aus dem Analyse-Teil aufgegriffen und mit der fertigen Anwendung und ihrer Architektur abgeglichen, also so was wie "die einzelnen Figuren und ihre Unterschiede, sind hier und hier da und da durch umgesetzt worden."

2.4.3 Überprüfung der Software mit Unit-Tests

Hier wird dann die breite der Unit-Tests gezeigt, deren Anzahl und die Beschränkungen, also Code-Abdeckung. Anschließend ein Ergebnis-Log von einem Lauf auf dem finalen Stand.

2.5 Fazit

3 Fazit und Ausblick

Die Ergebnisse wurden bisher nur dargestellt und erklärt und nicht bewertet und analysiert, dass soll hier geschehen unter "erreichte Ziele". Das Fazit soll Punkte zu Anwendbarkeit der Technologien, Entwicklung/Nachbau eines Spieleklassikers und Problempunkte, aber auch positives zu Support und Inbetriebnahme enthalten. Der Ausblick soll kommende Technologien und weitere Entwicklungspunkte für das Spiel beleuchten.

3.1 Fazit

3.1.1 erreichte Ziele

3.2 Ausblick

Hier kann auch noch in der bisher fehlende KI-Modus in Aussicht gestellt werden.

Literaturverzeichnis

- [1] BOWDEN KELLY: *TypeScript Node Starter*. <https://github.com/Microsoft/TypeScript-Node-Starter>. Version: 0.1.0, Abruf: 31.03.2018
- [2] CONTRIBUTORS, Mozilla: *MDN-Web-Dokumentation*. <https://developer.mozilla.org/de/>, Abruf: 31.03.2018
- [3] EXPRESSJS.COM CONTRIBUTORS: *express.js*. <https://expressjs.com/>. Version: 4.16.2, Abruf: 31.03.2018
- [4] [HTTPS://GITHUB.COM/MRDOOB](https://github.com/MRDOOB): *three*. <https://threejs.org/>. Version: 0.91.0, Abruf: 31.03.2018
- [5] [HTTPS://GITHUB.COM/SOCKETIO/SOCKET.IO/GRAPHS/CONTRIBUTORS](https://github.com/socketio/socket.io/graphs/contributors): *Socket.IO*. <https://socket.io/>. Version: 2.1.0, Abruf: 31.03.2018
- [6] IDC: *Absatz von Tablets, PCs und Smartphones weltweit von 2010 bis 2017 und Prognose für 2022 (in Millionen Stück)*. <https://de.statista.com/statistik/daten/studie/256337/umfrage/prognose-zum-weltweiten-absatz-von-tablets-pcs-und-smartphones/>, Abruf: 31.03.2018
- [7] MEDARCH: *The Secrets Of Archon*. <http://www.vintagecomputing.com/index.php/archives/44>. Version: 1.0, Abruf: 31.03.2018
- [8] MICROSOFT: *TypeScript - JavaScript that scales*. <https://www.typescriptlang.org/index.html>. Version: 2.8, Abruf: 31.03.2018
- [9] MICROSOFT: *Visual Studio Code*. <https://code.visualstudio.com/>, Abruf: 31.03.2018
- [10] NODE.JS FOUNDATION: *node.js*. <https://nodejs.org/en/>. Version: 9.4.0, Abruf: 31.03.2018
- [11] NYSTROM, Bob: *Game Programming Patterns*. 1. Auflage 2014. Genever Benning, 2014 <http://gameprogrammingpatterns.com/>. – ISBN 0990582906
- [12] THE JQUERY FOUNDATION: *jQuery - Webportal*. <http://jquery.com/>, Abruf: 31.03.2018
- [13] [HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/GAMES/ANATOMY\\$HISTORY](https://developer.mozilla.org/en-US/docs/Games/Anatomy$History): *Anatomy of a video game*. <https://developer.mozilla.org/en-US/docs/Games/Anatomy>. – is licensed under <https://creativecommons.org/licenses/by-sa/2.5/>
- [14] [HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/GAMES/TECHNIQUES/3D_ON_THE_WEB/BASIC_THEORY\\$History](https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/BASIC_THEORY$History): *Explaining basic 3D theory*. https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/Basic_theory. – is licensed under <https://creativecommons.org/licenses/by-sa/2.5/>

- [15] W3SCHOOLS, Refsnes Data: *W3Schools Online Web Tutorials*. <http://www.w3schools.com/>. Version: 2015, Abruf: 11.02.2015
- [16] WIKIPEDIA: *Archon (Computerspiel)*. [https://de.wikipedia.org/wiki/Archon_\(Computerspiel\)](https://de.wikipedia.org/wiki/Archon_(Computerspiel)). Version: Dezember 2014, Abruf: 31.03.2018

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Abkürzungsverzeichnis

Abb.	Abbildung
AJAX	ermöglicht asynchronen Datenaustausch mit z. B. Webservern (engl. für Asynchronous JavaScript and XML)
API	Programmschnittstelle nach aussen (engl. für Application Programming Interface)
ASCII	7-Bit Zeichencodierung (engl. für American Standard Code for Information Interchange)
AWL	Anweisungsliste, Assembler ähnlich
Code-Folding	logisch zusammengehörende Quelltextabschnitte werden in Abschnitte gruppiert, um diese einfach ein- bzw. auszublenden, erhöht die Lesbarkeit und Übersichtlichkeit
CPU	Elektronischer Rechner (engl. für Central Processing Unit)
CR	Wagenrücklauf (engl. Carriage Return)
CRC	zyklische Redundanzprüfung (engl. Cyclic Redundancy Check)
CSS	gestufte Gestaltungsbögen, legt die Darstellung des HTML Quellcodes im Browser fest (engl. für Cascading Style Sheets)
DB	Datenbaustein, Baustein zur Datenhaltung
DOM	Dokumentstruktur der Webseite (engl. für Document Object Model)
FB	Funktionsbaustein, wie FC nur mit Gedächtnis (Speicher in Form eines Datenbaustein
FC	Funktion
Field PG	Spezieller Laptop (Computer) für industrielle Umgebungen zum Programmieren einer Steuerung
FUP	Funktionsplan, Digitalen Logik Gattern ähnlich
GIF	Grafikaustausch Format, Animationsfähig (engl. für Graphics Interchange Format)
GUI	Grafisches Benutzer Interface (engl. für Graphical User Interface)
HAP	HTML Agility Pack, C#HTML Bibliothek zum verarbeiten von Webdokumenten
HMI	Mensch-Maschine Interface (engl. für Human Machine Interface)
HTML	Hypertext Auszeichnungssprache, HTML-Dateien sind die Grundlage des World Wide Web und werden von einem Webbrowser dargestellt (engl. für Hypertext Markup Language)

HTTP	Hypertext-Übertragungsprotokoll, Protokoll zur Übertragung von Daten über ein Netzwerk (engl. für Hypertext Transfer Protocol)
HTTPS	sicheres Hypertext-Übertragungsprotokoll s. a. HTTP (engl. für Hypertext Transfer Protocol Secure)
IDE	integrierte Entwicklungsumgebung (engl. für Integrated Development Environment)
Interface	Schnittstelle
IP	Internet Protocol
IPC	Industrie PC - Computer für Industriellen Einsatz
Java-Applet	Hilfsprogramm oder Tool, was in eine Webseite integriert wird
JavaScript	Skriptsprache, ursprünglich für dynamisches HTML in Webbrowsern entwickelt
JPEG	komprimierte Grafikdatei, auch JPG (engl. für Joint Photographic Expert Group)
JSON	kompaktes Datenformat zum Datenaustausch mit z. B. Webservern (engl. für JavaScript Object Notation)
KOP	Kontaktplan, Schaltplan ähnlich
LF	Zeilenvorschub (engl. Line Feed)
LRC	Längsparitätsprüfung (engl. Longitudinal Redundancy Check)
Mockup	Attrappe oder auch rudimentärer Prototyp (auch Maquette)
MSDN	Das Microsoft Entwickler Netzwerk (engl. für MicroSoft Developer Network)
MWSL	Mini Web Server Language, Serverbasierende Skriptsprache
o.V.	ohne Verfasser (bei Literaturverweisen)
OB	Organisationsbaustein
OS	das Betriebssystem (engl. für Operating System)
Parser	engl. to parse - „analysieren“ bzw. lateinisch pars - „Teil“ im Deutschen gelegentlich auch Zerteiler, Analysiert die Semantik des Scripts um daraufhin Aktionen durchzuführen
PC	Elektronischer Rechner (engl. für Personal Computer)
PG	Programmier Gerät, meist ein PC
PLC	s. a. SPS (engl. für Programmable logic controller)
PLCVarTab	Variablentabelle (Symboltabelle)
PN	s. a. Profinet
PNG	Grafikaustausch Format (engl. für Portable Network Graphics)

PROFINET	Bezeichnung für industriellen Netzwerkstandard (engl. für Process Field Network)
SCL	Strukturierter Text (engl. für Structured Control Language)
SDK	s. a. IDE (engl. für Software Development Kit)
SFB	System Funktionsbaustein
SFC	System Funktion
SOP	Same Origin Policy
SPS	Speicher Programmierbare Steuerung
ST	Strukturierter Text, s. a. SCL
SVG	skalierbare Vektorgrafik (engl. für Scalable Vector Graphics)
TCP	Transmission Control Protocol
TIA	Totally Integrated Automation
TN	Teilnehmer
UDT	Benutzerdefinierter Datentyp in Form einer Struktur (engl. für User Defined Typ)
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	Organisation zur Standardisierung von Webtechnologien (engl. für World Wide Web Consortium)
Webbrowser	auch kurz Browser (engl. to browse) steht für durchstöbern, abgrasen, durchsuchen - Software zum Darstellen von Daten, hauptsächlich Webseiten und deren Inhalt, können zu diesem Zweck mit Webservern kommunizieren
Webseite	s. a. HTML-Datei
Webserver	Ein Webserver speichert Webseiten und stellt diese zur Verfügung. Der Webserver ist eine Software, die Dokumente mit Hilfe standardisierter Übertragungsprotokolle (HTTP, HTTPS) an einen Webbrowser überträgt. In einer CPU mit PROFINET-Schnittstelle ist ein Webserver integriert, der mit anwenderdefinierten Webseiten erweiterbar ist
WPO	Webdaten Optimierung (engl. für Web Performance Optimization)
WWW	Internet (engl. für World Wide Web)
WYSIWYG	Man sieht im Editor sofort was man bekommt, sowohl textuell als auch grafisch dargestellt (engl. für what you see is what you get)

Stichwortverzeichnis

Das Verzeichnis ist in Haupt- und Unterbegriffe gegliedert. Ist ein Stichwort nicht unter den Hauptbegriffen gelistet, so ist es womöglich als Untereintrag zu finden.

Analyse, 5

Analyse neue Implementierung, 7

Architektur, 7

Aufbau, 3

Ausblick, 11

Erfüllung, 8

Fazit, 9, 11

Forschungsstand, 2

Hilfsmittel, 8

Hinführung, 1

Implementierung, 7, 8

Motivation, 1

Resultate, 8

Spiel Analyse, 5

Technologien, 8

Unittests, 9

Zentrale Begriffe, 2

Ziele, 2