

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 / Sperrvermerk	V
1 Einleitung	1
1.1 Hinführung	1
1.2 Aktueller Forschungsstand	1
1.3 Motivation	1
1.4 Ziele dieser Arbeit	1
1.5 Zentrale Begriffe	1
1.6 Aufbau	2
2 Hauptteil	3
2.1 Analyse	3
2.1.1 Analyse des klassischen Spiels	3
2.1.2 Anforderungsanalyse an neue Implementierung	3
2.2 Zusammenführung	3
2.2.1 Architektur	3
2.2.2 benötigte Technologien und Frameworks	3
2.3 Umsetzung	4
2.3.1 Hilfsmittel und Vereinfachungen	4
2.3.2 Schritte der Implementierung	4
2.4 Ergebnisse	4
2.4.1 fertige Architektur	4
2.4.2 Erfüllung der Anforderungen	4
2.4.3 Überprüfung der Software mit Unit-Tests	5
3 Fazit und Ausblick	7
3.1 Fazit	7
3.1.1 erreichte Ziele	7
3.2 Ausblick	7

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

Vorwort

Erklärung / Sperrvermerk

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 8. Februar 2018

Kevin Dyes

1 Einleitung

Die Einleitung soll hier nur in ihrer Struktur erklärt werden, die anderen Unterteile führen an das Thema heran, erklären es und beschreiben den Aufbau der restlichen Arbeit

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.

1.2 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.

1.3 Motivation

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.

1.4 Ziele dieser Arbeit

Hier soll bloß kurz und prägnant die Fragestellung genannt werden und eine kurze Analyse der einzelnen Worte des Titels stattfinden.

1.5 Zentrale Begriffe

?????

Hier fehlt
gendwie di
mal eine s
volle Über
tung/Reih
ge von Mo
tion zu Zie
zentrale B
fe.

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.

2 Hauptteil

2.1 Analyse

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.

2.1.1 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

2.1.2 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.

2.2 Zusammenführung

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

2.2.1 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.

2.2.2 benötigte Technologien und Frameworks

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

2.3 Umsetzung

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

2.3.1 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.

2.3.2 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 Ergebnisse

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 sowas 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.

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

Literaturverzeichnis

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, 3

Analyse neue Implementierung, 3

Architektur, 3

Aufbau, 2

Ausblick, 7

Erfüllung, 4

Ergebnisse, 4

Erklärung, Selbständigkeit, V

Fazit, 7

Forschungsstand, 1

Hilfsmittel, 4

Hinführung, 1

Implementierung, 4

Motivation, 1

Spiel Analyse, 3

Technologien, 3

Umsetzung, 4

Unittests, 5

Zentrale Begriffe, 1

Ziele, 1

Zusammenführung, 3