



DAS GROSSE SQL-SPIEL

THE SQL-ALCHEMIST

Software-Entwicklungspraktikum (SEP)
Sommersemester 2015

Angebot

Auftraggeber:
Technische Universität Braunschweig
Institut für Informationssysteme
Prof. Dr. Wolf-Tilo Balke
Mühlenpfordtstraße 23, 2.OG
D-38106 Braunschweig

Betreuer: Jan-Christoph Kalo

Auftragnehmer:

Name	E-Mail-Adresse
Gabriel Ahlers	g.ahlers@tu-braunschweig.de
Majid Dashtiepielehroud	m.dashtiepielehroud@tu-braunschweig.de
Ronja Friebe	r.friebe@tu-braunschweig.de
Stefan Hanisch	stefan.hanisch@tu-braunschweig.de
Fabio Luigi Mazzone	f.mazzone@tu-braunschweig.de
Nicole Naczki	n.naczki@tu-braunschweig.de
Denis Nagel	denis.nagel@tu-braunschweig.de
Luca Porcello	l.porcello@tu-braunschweig.de
Christian Reineke	c.reineke@tu-braunschweig.de
Christian Sander	christian.sander@tu-braunschweig.de
Carl Schiller	c.schiller@tu-braunschweig.de
Levent Muzaffer Üner	l.uener@tu-braunschweig.de
Sören van der Wall	s.van-der-wall@tu-braunschweig.de
Daniel Wolfram	d.wolfram@tu-braunschweig.de

Braunschweig, 24. April 2015

Inhaltsverzeichnis

1	Einleitung	4
1.1	Ziel	4
1.2	Motivation	4
2	Formale Grundlagen	6
3	Projektablauf	7
3.1	Meilensteine	7
3.2	Geplanter Ablauf	8
4	Projektumfang	10
4.1	Lieferumfang	10
4.2	Kostenplan	10
4.3	Funktionaler Umfang	10
5	Entwicklungsrichtlinien	12
5.1	Konfigurationsmanagement	12
5.2	Design- und Programmierrichtlinien	12
5.3	Verwendete Software	13
6	Projektorganisation	14
6.1	Schnittstelle zum Auftraggeber	14
6.2	Schnittstelle zu anderen Projekten	14
6.3	Interne Kommunikation	14
7	Glossar	15

Abbildungsverzeichnis

3.1	Gantt-Diagramm	9
-----	--------------------------	---

1 Einleitung

Im Folgenden wird zuerst das Motiv für das Projekt „Das Große SQL-Spiel“ erläutert und anschließend werden Projektablauf, -umfang und -organisation sowie sämtliche technische Grundlagen als auch Richtlinien geklärt. Dabei wird grundsätzlich zwischen einem Back- sowie einem Front-End unterschieden. Das Back-End steht hierbei für die Serverkomponente der Software. Das Front-End beschreibt die gesamte Oberfläche und Visualisierung der Software.

1.1 Ziel

Das Ziel ist es, eine plattformübergreifende, interaktive Spielesoftware zu entwickeln, um den Studenten der Pflichtveranstaltung „Relationale Datenbanksysteme I (RDBI)“ den Umgang mit der Datenbankanfragesprache SQL vorlesungsbegleitend spielerisch zu vermitteln. Sie soll ab dem Wintersemester 2015 vollständig in den Übungsbetrieb der Vorlesung integriert werden.

Es soll eine Möglichkeit geschaffen werden, praktische Aspekte im Bereich der Datenbankanfragen zu üben, da dies im Rahmen einer Lehrveranstaltung viele verschiedene Schwierigkeiten mit sich bringt. Zum Einen kann theoretisches Verständnis zwar gut vermittelt werden, jedoch ist es nur eingeschränkt möglich, genügend Übungsmaterial zur Verfügung zu stellen, damit eine komplexe Anfragesprache, wie SQL, geübt und somit auch verstanden werden kann.

Die Software soll genau diese Mängel aufgreifen und beheben.

1.2 Motivation

SQL hat die Fähigkeit komplexe, schwer nachvollziehbare Anfragen an eine Datenbank zu stellen. In der Theorie wird durchaus verstanden was hinter einer Datenbankanfrage steht, jedoch sind die Studenten meistens nicht in der Lage dies zügig in der Sprache SQL auszudrücken. Damit dieses verstanden werden kann, muss es permanent wiederholt und geübt werden. Zur Übung von SQL sind jedoch kaum sinnvolle Tools vorhanden, die ohne großen Aufwand benutzt werden können. Der Aufwand besteht darin, einen Datenbankserver zu installieren, zu konfigurieren, sich entsprechende Datenmodelle zu überlegen und diese mit sinnvollen Daten zu verknüpfen. Es soll eine Software entwickelt werden, die die Möglichkeit bietet, dieses den Studenten zu

erleichtern. Es wird eine komplett funktionsfähige Applikation erstellt, die unkompliziert aus jedem Webbrowser aufrufbar sein wird.

Nun bietet die zu entwickelnde Software genau diese Möglichkeiten. Wie ist es dann aber möglich die Studenten zu motivieren sich mit genau diesem Programm auseinander zu setzen um tatsächlich SQL zu üben? Hier greift das Prinzip der „Gamification“. Mit Hilfe von spielerischen Aspekten in Form eines Mini-Spiels und täglichen bzw. wöchentlichen Challenges sollen die Studenten motiviert werden. Durch immer wechselnde, zufällig generierte Herausforderungen und damit verbundenen Erfolgserlebnissen sollen die User belohnt werden und somit zum regelmäßigen Gebrauch dieser App animiert werden.

2 Formale Grundlagen

Das Back-End der Software wird in der Programmiersprache Java entwickelt. Zur Erstellung der grafischen Schnittstelle des Front-Ends wird die Skriptsprache JavaScript und die textbasierte Auszeichnungssprache HTML 5.0 verwendet. Diese Aspekte werden in Punkt 5, den Entwicklungsrichtlinien dieses Dokumentes, detaillierter aufgeführt.

Die in der Anwendung verwendete Sprache ist Englisch.

3 Projektablauf

3.1 Meilensteine

Alle wichtigen Ereignisse, wie zum Beispiel Abgabefristen die vom Kunden oder der Projektleitung gesetzt werden, aber auch intern im Team gesetzte Fristen, werden in Meilensteinen zusammengefasst und in einer Tabelle wiedergegeben. Dabei ist bei jedem einzelnen Meilenstein jeweils die Vervollständigung gemeint. Jedes Dokument wird vor jeder offiziellen Abgabe dem Betreuer zur Kontrolle vorgelegt.

Nummer	Meilenstein	Dokumente	Abgabetermin
1	Projektstart	-	13.04.15
2	Angebot an Betreuer	Angebot	20.04.15
3	Serverapplikation initialisiert	-	24.04.15
4	Angebot über Redmine	Angebot	24.04.15
5	Vervollständigen der Spielidee	-	24.04.15
6	Pflichtenheft an Betreuer	Pflichtenheft	06.05.15
7	GUI des Front-Ends	-	07.05.15
8	Userverwaltung	-	07.05.15
9	Pflichtenheft über Redmine	Pflichtenheft	13.05.15
10	Zwischenpräsentation	-	15.05.15
11	Fachentwurf an Betreuer	Fachentwurf	27.05.15
12	Fachentwurf über Redmine	Fachentwurf	03.06.15
13	Verwaltung für Benutzerrechte	-	24.06.15
14	Technischer Entwurf an Betreuer	Technischer Entwurf	24.06.15
15	Softwaretests	-	30.06.15
16	Technischer Entwurf über Redmine	Technischer Entwurf	01.07.15
17	Fertigstellung Minigame	-	24.06.15
18	Testdokumentation an Betreuer	Testdokumentation	08.07.15
19	Testdokumentation über Redmine	Testdokumentation	15.07.15
20	Präsentationsvorbereitung	-	20.07.15
21	Fertige Version	Quellcode, ...	23.07.15

3.2 Geplanter Ablauf

Ein Teil des Teams startet das Projekt mit dem Ausformulieren des Produktangebots. Währenddessen beginnt das Gestaltungsteam ihren kreativen Prozess. Ideenentwicklung und -austausch sowie Nützlichkeitsabwägung führen schnell zur fertigen Spielidee. Dieser Ablauf wird gleichzeitig genutzt, um zu prüfen welche Werkzeuge (Frameworks, Entwicklungsumgebungen, etc.) genutzt werden, und sich mit diesen vertraut zu machen. Dabei werden auch schon erste Grafiken wie etwa für den Start-Screen und das Hauptmenü entstehen.

Das Back-End-Team nutzt die Zeit um die Serverapplikation zu erstellen. Desweiteren werden Ideen für ein Administrationstool sowie zur User- und Datenverwaltung gesammelt und umsetzbar strukturiert. Zwischenzeitig wird ein Papierprototyp entstehen, der die Grundzüge der zu erstellenden Applikation enthält und das Team auf einen gemeinsamen Stand bringt.

Sobald der Kunde, sowie der Betreuer das Angebot angenommen haben, beginnt der praktische Teil.

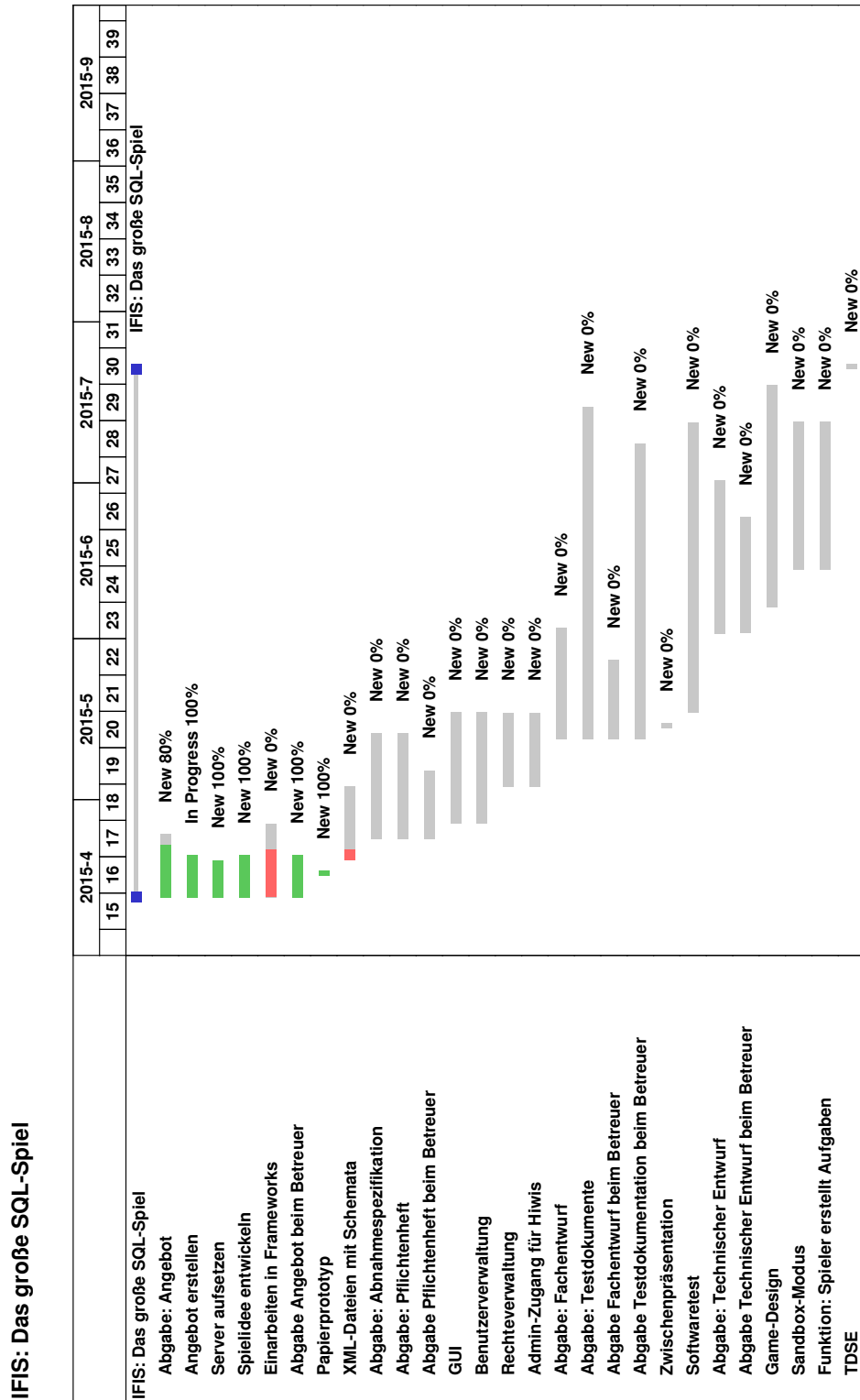
Das nächste Ziel wird sein, einen lauffähigen Prototypen zu entwickeln. Dafür erstellen ausgewählte Mitglieder des Entwicklerteams das sogenannte Pflichtenheft, um festzuhalten, was von dem Projektteam alles geliefert werden muss. Bis zur Fertigstellung des Prototypen und dessen Präsentation wird weiterhin ein Fachentwurf erstellt und eingereicht.

Das Front-End-Team wird nun sowohl das Hauptmenü als auch das „Alchemistenlabor“ (den eigentlichen „SQL-Vokabeltrainer“) bearbeiten, damit der repräsentative Teil des Projekts möglichst zur Zwischenpräsentation ausführbar ist. Parallel wird sich das Back-End-Team um die Nutzer- und Rechteverwaltung kümmern.

Nach der Zwischenpräsentation werden der Projektdokumentation noch der Technische Entwurf und die Testprotokolle hinzugefügt. Um die dazugehörigen Software-Tests kümmert sich das Team in der letzten Etappe.

Zum Abschluss des Projektes wird eine kleinere Gruppe aus ausgewählten Mitgliedern noch für eine gute Präsentation beim Tag der jungen Software-Entwickler (TDSE) sorgen.

Zur besseren Übersicht dieser einzelnen Abläufe ist ein Gantt-Diagramm erstellt worden.



21.04.2015

1/1

Abbildung 3.1: Gantt-Diagramm

4 Projektumfang

4.1 Lieferumfang

Zum Lieferumfang der Software gehören:

- der Quellcode des Programms
- die ausführbare Software
- die hinter der Nutzerschnittstelle liegende Datenbank
- das zugehörige Handbuch
- die Spezifikation und der Entwurf der Software
- die Protokolle der Tests

4.2 Kostenplan

Die für das Projekt anfallenden Kosten, werden auf ca. 316.000 Euro geschätzt. Davon entfallen 315.000 Euro auf den Lohn der Projektmitglieder (ausgehend von einem Arbeitslohn von 100 Euro pro Stunde und 30 Arbeitsstunden je Mitglied) und 1000 Euro auf den Kauf von Sprites, vorgefertigten Spielegerüsten und weiteren für das Spiel benötigten Objekten.

4.3 Funktionaler Umfang

Folgende Funktionen sollen in der abschließenden Version der Software enthalten sein:

1. Insgesamt soll das Spiel dem Nutzer 3, beziehungsweise 4 verschiedene Modi bieten:
 - a) Einen narrativen Modus, welcher dem Spieler ermöglicht in der Rolle des Alchemisten verschiedene Aufgaben zu erfüllen, um dann in der Geschichte des Spiels voranzuschreiten. Wobei es unabdingbar ist, dass der User SQL-Anfragen übt um in dem Modus weiter zu kommen.

- b) Einen Trivia-Modus, durch den die Spieler abseits eines festgelegten Handlungsstranges SQL-Abfragen üben können, indem sie zufällig gestellte Aufgaben lösen.
 - c) Einen Sandbox-Modus, der den Trivia-Modus durch vom Nutzer selbst erstellte Aufgaben erweitert. Diese Aufgaben stehen dann allen Nutzern zur Verfügung und können bewertet werden. Das Erstellen von Aufgaben jedoch bleibt freigeschalteten Spielern vorenthalten die sich im narrativen Modus bewährt haben. Dies wird durch das Back-End mit bestimmter Rechteverwaltung realisiert.
 - d) Einen Season-Modus, welcher von den Dozenten des Studienfaches RDBI genutzt werden kann, um die Software in den Übungsbetrieb zu integrieren.
2. Nutzerprofile und -statistiken, die den Fortschritt und die Erfolge eines jeden Spielers speichern. Außerdem sollen diese grafisch aufbereitet werden und von jedem User abrufbar sein.
 3. Ranglisten, über die sich die Nutzer untereinander vergleichen können um die Motivation zu fördern. Eventuell ist die Nutzung angemeldeten Spielern vorbehalten.
 4. Ein Minispiel, welches sowohl im Story-, als auch im Trivia-Modus zum Einsatz kommt. Dabei handelt es sich um ein „Jump & Run“-Spiel, in welchem die Spieler Gegenstände einsammeln um Punkte zu erhalten und dabei SQL-Hindernisse überwinden müssen.
 5. Eine Benutzerverwaltung, die einzelne Rechte, sowie Rollen verwalten soll.
 6. Ein Admin-Tool, damit Dozenten und deren Mitarbeiter die Software in ihre Übungen integrieren können. Insbesondere die Challenges, in dem Fall die Hausaufgaben, sollen hiermit erstellt und eingepflegt werden können. Beim Erstellen von Hausaufgaben kann man einen Titel auswählen und andere Eigenschaften festsetzen. Zum Beispiel kann ein Zeit-Slot gesetzt werden in der die Hausaufgabe bearbeitet werden muss.
 7. Auch die Studenten können im Anschluss der Bearbeitung dieses Tool nutzen um zu überprüfen ob sie die Hausaufgaben bestanden haben oder nicht.
 8. Das Back-End bietet ebenfalls eine Schnittstelle an, die mit dem Software-Modul des Teamprojektes kommuniziert und es mit entsprechenden XML-Schemata füllt.
 9. Die Studenten bekommen die Möglichkeit Kommentare zu den Hausaufgaben abzugeben. Somit bekommt auch der Betreuer der Hausaufgaben eine Review zum Schwierigkeitsgrad oder ähnlichen Kriterien. Auch dies wird durch das Back-End verwaltet.

5 Entwicklungsrichtlinien

5.1 Konfigurationsmanagement

Es gibt ein zentrales Repository auf Basis von Subversion (SVN), welches über das Redmine des Instituts für Softwaretechnik und Fahrzeuginformatik (ISF) zur Verfügung gestellt wird. Dieses Repository ist in drei Ordner unterteilt. In einem Ordner werden alle im Laufe des Projektes zu angefertigten Dokumente, die die Projektorganisation und den Kunden betreffen, abgelegt. Dazu zählen: ein Fachentwurf, eine Abnahmespezifikation, ein Pflichtenheft, ein technischer Entwurf, eine Testspezifikation und die entsprechenden Testprotokolle. Außerdem gibt es zwei weitere Ordner, in die alle Dateien der jeweiligen Teilprojekte, das Back- sowie das Front-End, abgelegt werden. Hierbei handelt es sich unter anderem um Dateien mit dem Quellcode der jeweiligen Software-Komponente.

Des Weiteren gibt es eine feste Kommentarstruktur, die bei jedem SVN-Commit, also dem Hochladen der geänderten Dateien in das SVN-Repository, eingehalten werden soll. Dies führt dazu, dass alle Änderungen nachvollziehbar, verständlich und vollständig sind. Um die Integrität der Applikation zu gewährleisten, soll ebenfalls nur vollständig ausführbarer Code in das Repository hochgeladen werden.

5.2 Design- und Programmierrichtlinien

Für die Entwicklung des gesamten Projektes ist festgelegt, dass sich die Entwickler an folgende im Rahmen der Veranstaltung „Software-Engineering I (SE I)“ vorgestellten Coding-Guidelines halten:

- **Einrückung:** Um den Code übersichtlicher zu gestalten, wird die Einrückungstiefe auf 4 Zeichen festgelegt.
- **Klammersetzung:** Jede Klammer die zur Schließung einer Methode bzw. einer Klasse gebraucht wird, wird in eine neue Zeile gesetzt.
- **Bezeichner:** Sämtliche Bezeichner (Parameter, Variablennamen, Methodennamen, Klassennamen, etc.) werden sinnvoll nach ihrem Inhalt benannt, damit direkt ersichtlich ist, was die jeweilige Methode berechnet oder welche Information in jeder Variable steht.

- CamelCaseNotation: Zur besseren Lesbarkeit des Quelltextes werden Methoden in CamelCaseNotation notiert.
- UPPER_CASE Notation: Alle Konstanten werden in Großbuchstaben, durch Unterstriche getrennt, geschrieben.
- lower_case Notation: Imports und Packages werden in Kleinbuchstaben geschrieben.

Darüber hinaus ist entschieden, sämtliche Kommentare auf Englisch zu verfassen. Das Software-Dokumentationswerkzeug Javadoc generiert parallel aus dem Java-Quellcode eine vollständige HTML-Dokumentationsdatei.

5.3 Verwendete Software

Als Entwicklungsumgebung wird IntelliJ IDEA von JetBrains verwendet, zum Erstellen von Diagrammen und technischen Zeichnungen das Visualisierungsprogramm Dia, sowie Microsoft Visio. Textdokumente, die das Projekt betreffen, werden mit Hilfe des Software-Pakets LaTeX angefertigt und formatiert.

Das Front-End wird mithilfe der HTML5 Spiele-Engine melonJS und der plattformunabhängigen JavaScript-Bibliothek jQuery entwickelt, das Admin-Tool des Back-Ends mit dem Open-Source-Framework AngularJS und dem CSS-Framework Bootstrap. Für die Server-Komponente des Back-Ends wird das Web-Framework play! verwendet.

Das Front-End-Team nutzt für das zu entwickelnde Mini-Spiel den Tiled Map Editor. Grafiken für die gesamte Software werden mit Hilfe des Bildbearbeitungsprogramm Adobe Photoshop CS6 entstehen.

Außerdem wird eine Software des mitarbeitenden Teamprojektes, die automatisch Schemata generiert zur Verfügung gestellt. Zudem kann auf eine Software, die im Rahmen einer Bachelorarbeit entwickelt wird, zurückgegriffen werden, die sich mit der automatischen Generierung von SQL-Statements und Aufgabenstellungen beschäftigt.

6 Projektorganisation

6.1 Schnittstelle zum Auftraggeber

Der Kunde steht den Entwicklern per Telefon-, E-Mail- und persönlicher Kommunikation zur Verfügung. Regelmäßige Meetings wurden direkt zu Anfang des Projektes auf donnerstags um 15 Uhr festgelegt.

6.2 Schnittstelle zu anderen Projekten

Die Mitglieder des externen Teamprojektes, die ihre Software zur Verfügung stellen, und der Verfasser der Bachelorarbeit, dessen Software ebenfalls zur Vergütung gestellt wird, können sowohl über die Projektbetreuer, als auch per E-Mail erreicht werden. Der Datenaustausch geschieht hierbei über GitHub.

6.3 Interne Kommunikation

Das SEP-Team kommuniziert über E-Mail bzw. Instant-Messenger und das eingerichtete Repository im Redmine. Außerdem finden sowohl mit den Betreuern und dem Kunden, als auch intern untereinander regelmäßige Treffen statt. Die Treffen werden mit dem Online-Termin-Planer Doodle organisiert. Zur Protokollierung dieser Meetings wird ein webbasierter Editor von Google verwendet.

7 Glossar

- Back-End: Das Back-End bezeichnet eine Schicht eines Programms. Das Back-End verarbeitet zur Verfügung gestellte Daten. Es entspricht hierbei der Serverkomponente des Programms.
- Front-End: Das Front-End ist das Gegenstück zu dem Back-End. Es beschreibt Ausgabe des Programms und ist für eine entsprechende Visualisierung zuständig.
- Javadoc: Javadoc ist ein Software-Dokumentationswerkzeug, das aus Java-Quelltexten automatisch HTML- Dokumentationsdateien erstellt.
- Redmine: Das Redmine ist eine webbasierte Organisations- und Projektverwaltungssoftware.
- Repository: Ein Repository ist ein verwaltetes Verzeichnis zur Speicherung von Daten
- Framework: Ein Framework ist ein vorgefertigtes Programmiergerüst.
- Spiele-Engine: Eine Spiel-Engine ist ein spezielles Spiele-Framework, das den Spielverlauf steuert und für die Visualisierung des Spiels sorgt.