Einleitung

Projektumfeld

Im Rahmen der Berufsausbildung zu Fachinformatikern für Anwendungsentwicklung wird in der Berufsschule, in diesem Fall die Berufsbildende Schule an der Brinkstraße, erwartet in Gruppenarbeit eine Projektdokumentation zu erstellen. Die Entwicklung der Applikation erfolgt dabei auf Laptops, mit dem Betriebssystem Microsoft Windows 10. Als Entwicklungsumgebung wurde der qtCreator gewählt, um die Applikation in C++ mit grafischen Elementen zu versehen und zu ergänzen.

Ausgangssituation

Ist-Zustand

Aktuell gibt es im Internet bereits eine Vielzahl an Tools für die Verwaltung von Charakterbögen des Tabletop-Roleplaying-Games (TTRPG) „Dungeons and Dragons“. Allerdings fehlt es an vielen dieser Tools hinsichtlich der Zugänglichkeit oder der spezifischen Funktionen. Außerdem haben viele jener Tools das Problem, dass sie nur mittels eines Abonnements zugänglich sind, auch wenn es vereinzelt Fälle gibt, in welchen die Nutzung mit einer einmaligen Gebühr verbunden ist.

Zeitgleich gibt es im Bereich der kostenlosen Werkzeuge und Programme jedoch häufiger den Fall das bestimmte Funktionen, wie das Hinzufügen von selbsterstellten Gegenständen (auch als „Homebrew“ benannt) nicht implementiert sind. Diese Funktion ist wiederrum bei semi-kostenpflichtigen (auch als Freemium bezeichnet) Modellen nur in der Premiumversion verfügbar, als direktes Beispiel lässt sich hier der Service von DnDBeyond nennen.

Projektziel

Das Ziel dieses Projekt ist es eine Software zur digitalen Verwaltung eines oder mehreren digitalen und interaktiven Charakterbögen für das TTRPG „Dungeons and Dragons“. Dazu zählen sowohl die Übersicht der Gesundheit in Lebenspunkten als auch die Kontrolle über des Inventar des jeweiligen Charakters. So soll es beispielsweise möglich sein, neue Gegenstände im Inventar abzulegen und auch die Ausrüstung des Charakters zu verwalten. Darüber hinaus soll es möglich sein, die verschiedenen Würfelproben, welche beim Spielen erforderlich sind zu simulieren und abzubilden, sowie die automatische Berechnung des Geldwertes in die verschiedenen Einheiten. Die fertige Software könnte kommerziell vertrieben werden, jedoch soll das Programm kostenlos verfügbar gemacht werden. Dabei muss jedoch beachtet werden, dass eine mögliche Kommerzialisierung von Wizards of the Coast, also dem Verlag, welcher das Copyright an Dungeons and Dragons besitzt, in der Vergangenheit bereits häufiger gegen solche Projekte juristisch vorgegangen ist. Zeitgleich gab es jedoch Fälle, in welchen der Verlag ähnliche Projekte gefördert hatte; die genauen Möglichkeiten der Kommerzialisierung sind also zum aktuellen Zeitpunkt nicht klar definierbar.

Ein weiteres Ziel ist das Erstellen einer Dokumentation im Rahmen der betrieblichen IHK-Abschlussarbeit, bzw. im Rahmer der schulischen Ausbildung.

Projektbegründung

Die Entwicklung dieses Projekts war vor allem dadurch gefordert, dass der organisatorische und materielle Aufwand, welcher oftmals mit einer Spielerunde, auch Session genannt, verbunden ist, sehr umständlich sein kann. Zu diesem Aufwand gehört unter anderem das Verwalten der verschiedenen Seiten Papier, die den Charakterbogen bilden, sowie die Würfel selbst. Da das Entwicklungsteam selbst aus begeisterten Spielen besteht, wurde entschlossen ein solches Tool als Projekt umzusetzen. Die eigenständige Entwicklung bietet zudem die Möglichkeit einige der gewünschten Optionen selbst zu verwalten und auch die Oberfläche anzupassen nach den persönlichen Wünschen hinsichtlich des Layouts.

Projektschnittstellen und Projektabgrenzungen

Die Anwendung soll dabei auf einem Desktopsystem unter Windows funktionsfähig sein. Eine Netzwerkverbindung wird nicht benötigt, das Programm ist nach dem Download und der Installation vollständig offline verfügbar. Die zugrundeliegende Sortierung der Daten innerhalb des Programms ist nicht Bestandteil dieses Projektes. Ebenfalls nicht Bestandteil dieses Projekt ist es, in der Grundversion verfügbaren Gegenstände abzubilden, zumal dies ein Verstoß gegen den Kopierschutz und die Lizenzbedingung von Wizards of The Coast darstellen würde.

Projektplanung

Projektphasen

Insgesamt standen für die Umsetzung des Projektes 80 Stunden zu Verfügung. Diese wurden in verschiedene Phasen aufgeteilt, die dadurch den gesamten Projektablauf abbilden. Gedrungen durch die Tatsache, dass diese 80 Stunden neben der regulären Arbeitszeit der Gruppenteilnehmer aufgebracht wurden, konnte das Projekt die Vollständigkeit nicht erreichen, zusätzlich erschwert durch häufige Krankheitsfälle. Darum sind viele der Komponenten noch unabhängig und nicht in einer ausführbaren Form entwickelt.

Die Planung für dieses Projekt begann am {xxxxxx}. Die Umsetzung des in dieser Dokumentation beschriebenen Projektteils fand neben der regulären Arbeitszeit im Zeitraum von {xx} bis {xx} statt. Die Dokumentation wurde vom {xx} bis zum {xx} erstellt. Aufgrund anderer Verpflichtungen und Termine wurden die geplanten 80 Stunden über einen längeren Zeitraum verteilt und nicht auf einmal abgearbeitet. Die grobe Zeitplanung aller Hauptphasen kann Tabelle 1 entnommen werden. Eine detaillierte Zeitplanung aller Projektphasen und der jeweiligen Aufgaben befindet sich im Anhang.

Abweichungen vom Projektantrag

Anders als im Projektantrag geplant musste das Entwicklungsteam einige Abstriche machen. Dazu zählten untere anderem die Möglichkeit zwischen den Charakterbögen zu wechseln, bzw. diese separat zueinander abzuspeichern. Nach weitere Prüfung wurde außerdem auf das Bereitstellen des Downloads über eine GitHub-Webpage verzichtet, da diese nicht kostenlos zur Verfügung standen, wie angenommen.

Ressourcenplanung

Für die Umsetzung des Projekts waren verschiedene Ressourcen erforderlich, die grob in Hardware-, Software- und Personalressourcen unterteilt werden können Die notwendige Hardware bestand aus Laptops, Arbeitsplätzen und Besprechungsräume für die Absprache, diese wurden entweder durch die Räumlichkeiten der Berufsschule, oder über Internetdienste wie beispielsweise Discord realisiert.

Bei der genutzten Software wurde darauf geachtet, dass keine zusätzlichen Kosten durch Lizenzen entstehen, da kein Budget für die Entwicklung vorhergesehen wurde. Deswegen wurde hauptsätzlich auf freie Software oder Software mit bereits vorhandenen Lizenzen verwendet. Zur Implementierung wurde die Entwicklungsumgebung qtCreator verwendet, für die eine kostenlose, OpenSource Lizenz verfügbar ist. Die grafischen Komponenten wurden mit einer lizensierten Version von Adobe XD im Rahmen der Adobe Creative Cloud erstellt. Die Versionsverwaltung erfolgte über Git, über den Service von Github. Die Projektdokumentation wurde mit Microsoft Word erstellt und mit Grafiken von draw.io erweitert. Eine genaue Auflistung der Ressourcen findet sich im Anhang wieder.

Projektplanung

Im Rahmen der Planung des Projekts wurden zunächst die benötigen Elemente definiert, wodurch sich die folgenden Aufgaben ergaben:

Analyse:

* Prüfen der Anforderungen
* Durchführung einer Ist-Analyse

Projektplanung:

* Erstellung der Leistungsbeschreibung
* Erstellung einer visuellen Vorlage

Implementierung:

* Vorbereitung der Entwicklungsumgebung
* Implementierung einer funktionsfähigen Oberfläche
* Erstellung eines HMI (Human-Machine-Interface)
* Implementierung der Speicherfunktion
* Implementierung der Würfelsimulation und Logik
* Implementierung der allgemeinen Berechnungslogik

Testen:

* Testen durch das Entwicklungsteam im Rahmen eines Blackbox-Tests

Abnahme und Einführung:

* Weitere Tests anhand der Leistungsbeschreibung
* Bereitstellung des Programms als Releaseversion über GitHub

Dokumentation:

* Erstellung der Nutzungsanleitung
* Erstellung der Entwicklerdokumentation
* Erstellung der Projektdokumentation

Zeitplanung

Die Umsetzung des Projekts erfolgt in der außerberuflichen Zeit der drei Projektteilnehmer über einen Zeitraum von 80 Stunden. Dabei verteilen sich die 80 Stunden auf die Aufgaben wie folgt:

Analyse: 3h

Projektplanung: 13h

Implementierung: 27h

Testen: 4h

Abnahme und Einführung: 4h

Dokumentation: 29h

Kostenplanung

Es wurden keine Kosten eingeplant und entsprechend entfällt auch die Notwendigkeit einer definierten Kostenplanung. Ebenso entfällt eine Amortisierung, da die Applikation nicht kommerziell vertrieben werden soll.

Durchführung

Ist-Analyse

Für das TTRPG „Dungeons and Dragons“ [©Wizards of the Coast] benötigen Spieler:Innen einen sogenannten Charakterbogen. Dieser stellt die, für die Spieler:Innen relevanten Informationen dar, um ihren jeweiligen eigenen Charakter zu spielen. Diese werden traditionell auf Papier gedruckt und beschrieben, also eine rein analoge Ansicht der Daten, weshalb es sich anbietet diese reine Informationsanzeige in die digitale Form umzuwandeln. Darüber hinaus lassen sich mit einer digitalisierten Form auch einige der spielrelevanten Prozesse, wie beispielsweise das Würfeln oder die Verwaltung des Inventars eines Charakters, wodurch sich der Kauf von zusätzlichen Materialien wie beispielsweise Würfeln, erübrigt. Zudem vereinfacht die digitale Version das Spielen mit anderen Personen, da zum Beispiel der Erzähler des Spiels (Allgemein als Dungeonmaster / DM benannt) die Charakterbögen benötigt, um die Kampagnen sinnvoll vorzubereiten. Durch eine digitale Form kann das Teilen der Dateien dann z.B. über E-Mail ablaufen.

Projektplan

A

Entwurfsphase

In dieser Phase wurden mittels Adobe XD, sowie einiger, bereits existierende digitale Charakterbögen als Inspiration, die verschiedenen grafischen Komponenten erstellt. Dabei wurde der erste Entwurf des zugrundeliegendem Layouts gemeinsam auf Papier erstellt und anschließend in Adobe XD umgesetzte, da dort die Möglichkeit besteht die einzelnen Elemente als SVG-Datei zu exportieren, welche im Anschluss im qtCreator als grafische Oberflächen eingepflegt werden können. Da bereits im Vorfeld die Entscheidung getroffen war, das Projekt in C++ umzusetzen war einer der Anforderung des Designs eine gewisse Modularität zu ermöglichen und so den Umgang, bzw. die Arbeit mit dem qtCreator zu vereinfachen.

Gesamtprojekt

Für die Implementation und Integration der GUI konnten die in Schritt {X} erstellten, grafischen Komponenten in qtCreator eingefügt werden, um so das Layout zu definieren und mit Funktionen zu befüllen. Auf der technischen Seite werden die Informationen, die in dem Programm eingepflegt und verwaltet werden, in Form eines JSON-Objekts gespeichert (Anhang Nr. {x} zeigt ein Beispiel eines solchen JSON-Objekts). Der Vorteil dieser Speichermethode liegt vor allem darin, dass selbst ohne das Programm es möglich ist, die Werte einzusehen. Darüber hinaus ist das JSON-Format ein bereits etabliertes Modell, welches auch häufig für APIs verwendet wird, weshalb es in der Zukunft mit dem Format auch möglich sein kann, die Charaktere über einen Login mittels einer API abzurufen.

Durch die Anforderungen der Berechnung von Werten, sowie der Ausgabe von Würfelproben wird zusätzlich zu der reinen Informationsanzeige auf einiges an mathematischer Logik, in Form von einzelnen Komponenten benötigt.

Logikkomponenten

Zentral zum Projekt gibt es vier separate Logikkomponenten, welche sich in ihrer Komplexität zum Teil sehr unterscheiden. Die simpelste Komponente ist dabei die Darstellung des Geldwertes. Dungeons and Dragons verwendet dabei vier, bzw. fünf Arten der Währung, welche alle untereinander verrechnet werden können. Diese Währungen sind Kupfer-, Silber-, Gold- und Platinmünzen. Je nach Präferenz gibt es auch noch die Möglichkeit eine fünfte Währung zu verwenden, Electrum, welche sich von ihrer Wertigkeit zwischen Silber- und Goldmünzen befindet. Die Umrechnung basiert dabei darauf, dass alles in Zehnerschritten erfolgt. So entsprechen beispielsweise zehn Silbermünzen einer Gold-, bzw. 100 Kupfermünzen (oder 100 Electrum- und 1.000 Kupfermünzen).

Die zweite Komponente reguliert die Belastung der Charaktere. Ein Charakter kann nur ein bestimmtes Maß an Ausrüstung mit sich führen. Die Ermittlung, wie viel der jeweilige Charakter tragen kann wird dabei durch eine Regel des Spiels bestimmt. Sollte ein Charakter sich dazu entscheiden, mehr zu tragen, als er oder sie fähig ist, so erleidet der Charakter eine Verschlechterung, beispielsweise bei Würfelproben. Sollte der Charakter zudem vollständig „überladen“ sein, also deutlich mehr tragen, als spieltechnisch erlaubt ist, so erleidet er weitere Nachteile, wie beispielsweise eine weitere Reduktion der Geschwindigkeit.

Die dritte Komponente ist die Logik des Würfelns. Für eine normales Spiel werden eine Vielzahl von Würfeln benötigt, einige von ihnen besitzen meist weitere Modifikatoren. Einer dieser Modifikatoren ist dabei das Prinzip des Vor- bzw. Nachteils. Hat ein:e Spieler:In Vor- bzw. Nachteil werden so beispielsweise zwei, statt ein Würfel geworfen. Im Falle des Vorteils wird dann der höhere der beiden Ergebnisse für das Spiel verwendet, bei einem Nachteil der niedrigere Wert. Außerdem haben viele der Würfelproben auch mathematische Modifikatoren, welche sich aus den bestimmten Werten der Charaktere ergibt.

Entsprechend muss die Würfellogik eben jene Modifikatoren, sowie die Möglichkeit eines Vor- und Nachteilwurfes ermöglichen. Außerdem sollen die Spieler:Innen in der Lage sein die Würfellogik zu verwenden, ohne einen direkten Modifikator auszuwählen, da der/die Leiter:In des Spiels auch zusätzliche Würfe anfordern kann, um beispielsweise einfach nur eine „Probe“ auf das Glück des Charakters zu machen.

Die komplexeste der Komponenten füllt automatisch einige der Werte auf, die sich aus den, von den Spielern ausgewählten Attributswerten ergeben. Da viele der Felder eine Sonderregel und weitere Modifikatoren beinhalten, wird dabei eine Vielzahl von Funktionen benötigt, einige davon verweisen zusätzlich noch auf die Würfellogik und besitzen ihre eigenen Würfelmodifikatoren.

Implementierungsphase

Zunächst wurde auf den Geräten der beteiligten Personen die Community Version der qtCreator Oberfläche, sowie die dazugehörigen Module installiert. Anschließend wurde ein Grundlegendes Projekt, sowie ein dazugehöriges GitHub-Repository angelegt. Die Aufgaben wurden dabei innerhalb der Gruppe folgendermaßen verteilt:

Erarbeitung der Designvorlage: Pascal Asbrock

Grafische Umsetzung in Adobe XD: Pascal Asbrock

Da die Implementierung des Programms im Rahmen der qtCreator Umgebung stattfand, war es zunächst notwendig sich mit der Entwicklungsumgebung vertraut zu machen, sowie mit der, im Vorfeld festgelegten Programmiersprache C++. Durch die Entscheidung, die grafischen Elemente im Vorfeld als SVG-Dateien zu rendern, erübrigt sich die kleinteilige Einrichtung der einzelnen Interaktionselemente in der Oberfläche von qtCreator, wodurch wiederrum Zeit effizienter genutzt werden kann.

Da die fertige Applikation darauf ausgelegt ist, ausschließlich auf Windows-Desktopsystem ausgeführt zu werden, ist es das Prüfen der Kompatibilität bzw. der Portierung auf andere Betriebssysteme nicht nötig und beschränkt sich auf die Kompatibilität für verschiedene Windowsversionen (bspw. Windows 10 [Home | Professional | Enterprise]).

Entwicklung der Logikkomponenten

Die Entwicklung der Logikkomponenten erfolgte parallel zueinander und wurde in separaten Dateien realisiert. Die Würfellogik besteht dabei im Kern aus einigen Funktionen welche als Zufallsgenerator agieren und anschließend mit den, in Form von Parametern übergebenen, Modifikationswerten verarbeitet werden.

Um die Funktionsweise der Würfellogik zu erklären, schauen wir uns die Funktion D20Roll an, welche einen 20-seitigen Würfel simuliert:

Zunächst wird ein allgemeiner Zufallsgenerator, basierend auf der Systemzeit initialisiert. Auch wenn diese Form keinen „echten“ Zufall simuliert, reicht die Zufallsrate im Rahmen des Projekts aus.



Anschließend werden die mathematischen Limits der Funktion festgelegt, in unserem Beispiel als 1-20



Das direkte Würfelergebnis wird dann als Variable gespeichert,



und im Anschluss mit den Modifikatoren verändert. Der Standardwert für den Fall, dass kein Modifikator übergeben wurde ist dabei 0:

A black background with white text

Description automatically generated

Letztendlich wird das Ergebnis berechnet und an die Oberfläche zurückgegeben

A black screen with yellow text

Description automatically generated

Die einfachste Komponente, die Verrechnung des Geldes, besteht im Kern aus zwei Funktionen, welche über zwei Parameter bestimmt welche Art von Münze verwendet, bzw. gefordert wurde und der Betrag. Um die Rechnung zu vereinfachen, erfolgt die Verarbeitung auf Basis der kleinsten Einheit, also der Kupfermünze. Der entgegengenommene Wert wird dabei aufgeteilt und als Wert für Kupfermünzen gespeichert, bei der Rückrechnung wird der Wert als Kupfermünzen mit dem Wert Zehn verrechnet, um so die jeweilige Anzahl an Münzen zu erreichen.

Unterdessen erfolgt die Berechnung der Belastung / Tragkraft. Dabei wird zunächst der Wert aus dem Feld ausgelesen, welcher zu Beginn die Belastung bestimmt. Dadurch spart sich das Programm die wiederholte Berechnung. Die Berechnung erfolgt dabei nach den etablierten Regeln des Spiels, also nach der folgenden Staffelung:

Leichte Belastung = Wert der Stärke \* 5  
Mittlere Belastung = Wert der Stärke \* 10  
Maximale Belastung = Wert der Stärke \* 15

Die vierte und wichtigste Komponente, die Logik der Attribute, besteht dabei aus {x} Funktionen. Jede dieser Funktionen erhält als Parameter den jeweiligen Basiswert, welcher zu der jeweiligen Fähigkeit gehört. Eine Auflistung aller Relationen befindet sich im Anhang. Die Funktionen werden dabei beim Starten des Programms ausgeführt, falls es im JSON-Objekt keinen Wert gibt.

Testen

Das Testen der Applikation sollte im Rahmen eines gemeinschaftlichen Spielrunde passieren, durch die Nicht-Fertigstellung des Programms konnte dieser Schritt jedoch nicht erreicht werden. Neben diesem geplanten Testfall war es außerdem vorgesehen die Komponenten einzeln zu testen, was auch erfolgreich war.

Abnahmephase

Im Gegensatz zu herkömmlichen Programmierprojekten gibt es aufgrund der Umstände in unserem Fall keine klassische Abnahmephase. Stattdessen erfolgt, nach dem Testen, lediglich eine Einführung, bzw. eine Veröffentlichung. Die Veröffentlichung sollte dabei vollständig über das, auf GitHub gelagerte Repository erfolgen, mit der Bedienungsanleitung als eigene Datei im Repository. Der Download erfolgt über die frei zugängliche Schnittstelle von GitHub.

Mögliche Weiterentwicklungen der Applikation

Eine der möglichen Weiterentwicklungen könnte es sein eine optionale Netzwerkschnittstelle zu erstellen, welche das Synchronisieren der Charakterbögen auf mehreren Geräten gleichzeitig zu ermöglichen. Dabei würde ein Webserver sowohl die Synchronität innerhalb eines Benutzerkontextes als auch das Erstellen von Gruppen durch den DM, welcher dann ebenfalls die Charakterbögen einsehen könnte.

Kundendokumentation

Der Download, sowie die Bedienungsanleitung, wird vollständig über GitHub bereitgestellt. Das Anlegen der Bedienungsanleitung erfolgt dabei nach Fertigstellung der Applikation, um so sicherzustellen, dass die dort beschriebenen Funktionen und die generelle Funktionsweise der Releaseversion entspricht.