Abschlussarbeit Sommer 2016

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Charaktersteuerung

Programmierung einer intuitiven Charaktersteuerung im Projekt „Sisyfox“

Abgabetermin: 26.04.2016

Prüfungsbewerber:
Marvin Jung
Weddinger Str. 23
38690 Immenrode

Ausbildungsbetrieb:
ckc ag
Am Alten Bahnhof 13
38122 Braunschweig

Inhaltsverzeichnis

Seite

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Die ckc group ist ein von Christian Krentel im Jahr 1989 gegründeter IT- und Business-Consulting-Anbieter in Braunschweig. Sie gehört in dem Bereich zu den führenden Unternehmen Deutschlands. In den Standorten Braunschweig, Berlin, Darmstadt, Hamburg, Dortmund und München beschäftigt sie rund 400 Mitarbeiter[1].

Hauptbranchen des Unternehmens sind die Automobilindustrie samt Zulieferer sowie Banken, Versicherungen, Luft- und Raumfahrt, Retail, Transport und Logistik. Die Kernkompetenz von ckc liegt im Bereich der IT nicht nur auf der Softwareentwicklung, sondern zu Teilen zusätzlich auf der Managementberatung.

1.1. Projektbeschreibung

Das "Sisyfox" Projekt wird entworfen, um für firmeninterne Events (z.B. Seminare, Konferenzen, Sommerfeste, etc.) eine Attraktion zu bieten, die nicht nur mit Spaß verbunden wird, sondern aus der sich auch ein Lernwert ergibt. Das Spiel stellt in einer vereinfachten Form den Mythos des Sisyphos und sein immerwährendes Scheitern gamifiziert[2] dar. Den Mitarbeitern soll dabei der Umgang mit dem Erlebnis des Scheiterns näher gebracht werden. Außerdem soll diese Gamifizierung zu einer Motivationssteigerung führen.

Der Charakter im Spiel wird mit einem Aufbau, auf dem ein 1.20m großer Ball als Steuerkugel[3] befestigt ist, vom Spieler gelenkt. Dafür steht der Aufbau vor dem Spieler auf dem Boden. Die Kugel kann durch den Aufbau mit den Händen auf der Stelle bewegt werden. Die Sensordaten des Aufbaues werden direkt an den Computer übermittelt und müssen in der Steuerung umgesetzt werden.

Meine Aufgabe in dem Projekt besteht darin, die Steuerung für den Charakter im Spiel zu programmieren. Diese wird in der Programmiersprache C# geschrieben, damit sie in der Spieleentwicklungsumgebung "Unity" als Skript in das vorhandene Projekt eingebunden werden kann. Hierbei muss auf eine intuitive Steuerung geachtet werden, da die Attraktion einer großen Zielgruppe zur Verfügung gestellt werden soll. Im Projekt vorhanden sind bereits sämtliche Modelle, Animationen, sowie Sounds und Hintergrundmusik, aber auch schon einzelne, von mir früher programmierte Skripte, die sich z.B. um das Benutzerinterface kümmern. Die benötigten Daten für die Steuerung des Charakters werden von einer Trackballmaus zum Computer übertragen und müssen dort in die Bewegung umgesetzt werden. Außerdem muss im Rahmen der Projektarbeit bestimmt werden, wie sich der Charakter in der virtuellen Welt verhält, sowie in welchen Situationen der Spieler das Spiel vorzeitig verlieren kann und ab welchem Zeitpunkt das Spiel gewonnen ist.

[1] *Unternehmensporträt der ckc.* [2016] [1]
[2] *Gamifizierung Definition.* [2016] [2]
[3] Wie in einem Kugellager.

1.2. Projektziel

Mit dem Projekt möchte ich eine gewisse Hartnäckigkeit, gefolgt von gesteigerter Motivation durch Erfolgserlebnisse bei den Mitarbeitern, sowie Besuchern von Firmenfesten und anderen Events erzielen. Der Benutzer wird mit jeder gespielten Runde vor die Aufgabe gestellt eine scheinbar unerreichbare Hürde zu überwinden. Er wird viele Versuche brauchen, um den Gipfel zu erreichen. Der spielerische Aspekt sorgt unweigerlich dafür, dass der Benutzer leichter mit häufigen Niederlagen umgehen kann, um anschließend den anfangs weit entfernten Berg erneut herauszufordern. Sobald der Spieler am Gipfel angekommen ist, gibt es auditive und visuelle Erfolgsmeldungen. Dazu kommt eine Anzeige, wie lange der Weg zur Bergspitze gedauert hat, damit der Spieler sich mit anderen messen kann und ein Wettkampfgefühl entsteht.

1.3. Projektabgrenzung

Im Projekt sind bereits ein Spielehauptcontroller, sowie ein Animations- und Bewegungssteuerung vorhanden. Der Hauptcontroller sorgt für die Verwaltung der Grundstatus[4] im Spiel. Außerdem speichert er die wichtigsten Variabeln, wie unter anderem Spieldauer oder die Höhe abweichend vom Fuß des Berges. Es wird lediglich ein Verweis via Assoziation in der Charaktersteuerung eingebunden.

[4] Die vier Status sind in aktiv, spielend, gewonnen und verloren.

2. Projektplanung

2.1. Projektphasen

Für die Umsetzung des Teils des Projekts stehen insgesamt 70 Stunden zur Verfügung. Die Softwareentwicklung wurde vor Beginn des Projekts auf die verschiedene Projektphasen aufgeteilt. In der folgenden Tabelle [1] lässt sich die grobe Zeitplanung ablesen.

Projektphase	**Geplante Zeit**
Ist-Analyse	6 Std
Soll-Konzept	11 Std
Implementierung	24 Std
Integration in das Unityprojekt	8 Std
Testen und Fehlerbehebung	11 Std
Dokumentation	10 Std
Gesamt	**70 Std**

Tabelle 1.: Grobe Zeitplanung

Eine verfeinerte Version, in der die einzelnen Hauptphasen noch untergliedert sind, befindet sich im Anhang *Anhangsverweis einfügen* auf der Seite *Seitenzahl einfügen*.

2.2. Ressourcenplanung

3. Analysephase

3.1. Ist-Analyse

In dem Spiel gibt es viele kleine Details, welche über das Einbinden von Scripten und das Platzieren von 3D-Modellen realisiert werden. Dazu gehört zunächst die Spielkarte auf der sich der Charakter bewegt. Diese wurde von anderen Teammitgliedern erstellt, mit 3D-Modellen wie z.B. Bäumen, Sträuchern oder Gesteinsbrocken bestückt und in das Projekt implementiert. Zusätzlich wurden 3D-Modelle von Wolken und Vögeln erstellt, die sich im Kreis um den Hauptberg bewegen[1]. Der Spielecharakter namens „Sisyfox“ gehört zu den schon erstellten 3D-Modellen und wurde ebenfalls schon mit einer vollständiger Animation versehen. Wie in Sektion 1.3 (Projektabgrenzung) schon angedeutet ist der Hauptcontroller, als einer von vielen Scripten, schon vorhanden. Weitere wichtige Scripte sind die Kamerasteuerung, sämtliche Animationsscripte, eine Höhenmeteranzeige und ein Arduinokommunikationsscript, welcher eine Verbindung zu einer Hardwarekonstruktion namens „Arduino[2]“ aufbaut. Letzter sorgt dafür, dass sich relativ zur Höhe auf der sich der Charakter befindet, ein angeschlossener Ventilator stärker dreht und eine Nebelmaschine kurz vor Erreichen des Gipfels anspringt. Um das Spiel zu komplettieren fehlt nun die Charaktersteuerung.

3.2. Anwendungsfälle

[1] Die Bewegung wurde mit einem Script realisiert.
[2] *Arduino Beschreibung.* [2016] [3]

4. Entwurfsphase

4.1. Zielplattform

Zur Auswahl standen die IDEs[1] „Unity 3D Engine“ und „Unreal Engine 4“. Gewählt wurde die Unity Engine, da es in der Unreal Engine mangels grundlegender Vorkenntnisse zu einer deutlich längeren Entwicklungszeit kommen würde. Es müsste zunächst ein großer Teil der Zeit für Einarbeitung aufgewandt werden. Ein weiterer Grund ist das sehr einfache und intuitive Einbinden von Gameobjekt[2]-Scripten.

4.2. Architekturdesign

Für den Architekturaufbau standen das „Model-View-Controller-“ (MVC) und das „Presentation-Abstraction-Control-Muster“ (PAC) zur Auswahl.

Des MVC Muster (Siehe Schema Abbildung A.5) besteht grundsätzlich aus drei Strukturen, die unterschiedliche Ebenen im Programmaufbau darstellen. Diese unterteilen sich in „Modell-“, „View-“ und „Controller-Ebene“. Die Unterteilung dient hierbei dem Zweck, dass jede Ebene flexibel und unabhängig von den Anderen verändert, ersetzt oder erweitert werden kann. Das Modell beinhaltet die Daten, die für die Anzeige nötig sind. Über die View-Ebene werden die Daten aus dem Modell dem Benutzer angezeigt und des Weiteren mögliche Benutzereingaben entgegengenommen. Wenn sich Daten im Modell ändern wird im Regelfall die View-Ebene mithilfe eines Beobachters[3] benachrichtigt, sodass sich die View aktualisieren kann. Die entgegengenommen Benutzereingaben werden von dem Controller übernommen und verarbeitet. In dem objektorientierten Ansatz führt der Controller Methoden im Modell aus, damit die Aktionen des Benutzers wirksam werden. Außerdem kann der Controller die View direkt manipulieren, um z.B. die angezeigten Eingabemöglichkeiten zu verändern.

Um ein System in verschiedenen Einzelteilen zu erstellen, benötigt man ein Muster, dass jede Aufgabe in einem anderen Teil des System abbildet. Dadurch wird hohe Flexibilität ermöglicht, die sich auch im Wartungsaufwand widerspiegelt. Im PAC Muster (Siehe Schema Abbildung A.6) teilt man das System in zwei Richtungen auf, zunächst in drei Einheiten, die grafische Oberfläche, einer Kommunikationsschnittstelle und einem Datenmodell. Diese Einheiten ähneln dem MVC-Muster.
Darüber hinaus teilt sich das Architekturmuster hierarchisch auf sogenannte „Agenten“ auf. Diese Agenten werden auf drei Schichten verteilt, auf *Top-Level-*, *Intermediate-Level-* und *Bottom-Level*-Agenten. Den *Top-Level*-Agenten gibt es nur einmal im System, er übernimmt die globalen Aufgaben. Für die *Bottom-Level*-Agenten ist eine möglichst in sich abgeschlossene Aufgabe vorgesehen, die über die *Intermediate-Level*-Agenten an das *Top-Level* kommuniziert werden.

[1] eng: integrated development environment, de: Integrierte Entwicklungsumgebung
[2] Gameobjekte sind die Grundbausteine von jedem Unity Spiel.
[3] Der Beobachter meldet sich bei der View-Ebene, sobald sich Daten im Modell ändern.

4.3. Entwurf der Benutzeroberfläche

4.4. Geschäftslogik

4.5. Maßnahmen zu Qualitätssicherung

5. Implementierungsphase

5.1. Implementierung der Benutzeroberfläche

5.2. Implementierung der Geschäftslogik

6. Abnahme- und Einführungsphase

6.1. Abnahme

6.2. Einführung

6.3. Dokumentation

7. Fazit

7.1. Soll-/Ist-Vergleich

7.2. Lessons Learned

7.3. Ausblick

Literaturverzeichnis

[1] *Unternehmensporträt der ckc.* 2016. URL: http://www.ckc-group.de/index.php?id=709.

[2] *Gamifizierung ist die Übertragung von spieltypischen Elementen und Vorgängen in spielfremde Zusammenhänge mit dem Ziel der Verhaltensänderung und Motivationssteigerung bei Anwenderinnen und Anwendern.* 2016. URL: http://wirtschaftslexikon.gabler.de/Definition/gamification.html.

[3] *Die Arduino Hardware ist ein Entwicklerboard zum Basteln von eigenen Schaltkreisen und zum Ausführen von Steuerungsprogrammen für die Verwaltung von angeschlossener Elektronik.* 2016. URL: https://www.arduino.cc/en/Guide/Introduction.

A. Anhang

Abbildung A.1.: Benutzer Oberfläche Skizze

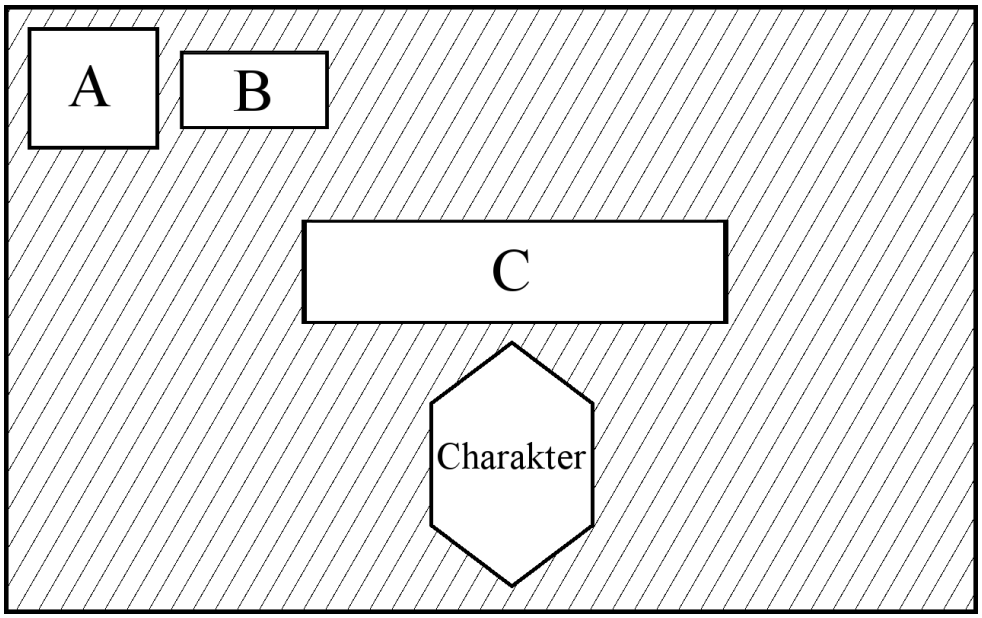

Abbildung A.2.: Benutzer Oberfläche Im Spiel

Abbildung A.3.: Zustandsdiagramm Statusverwaltung

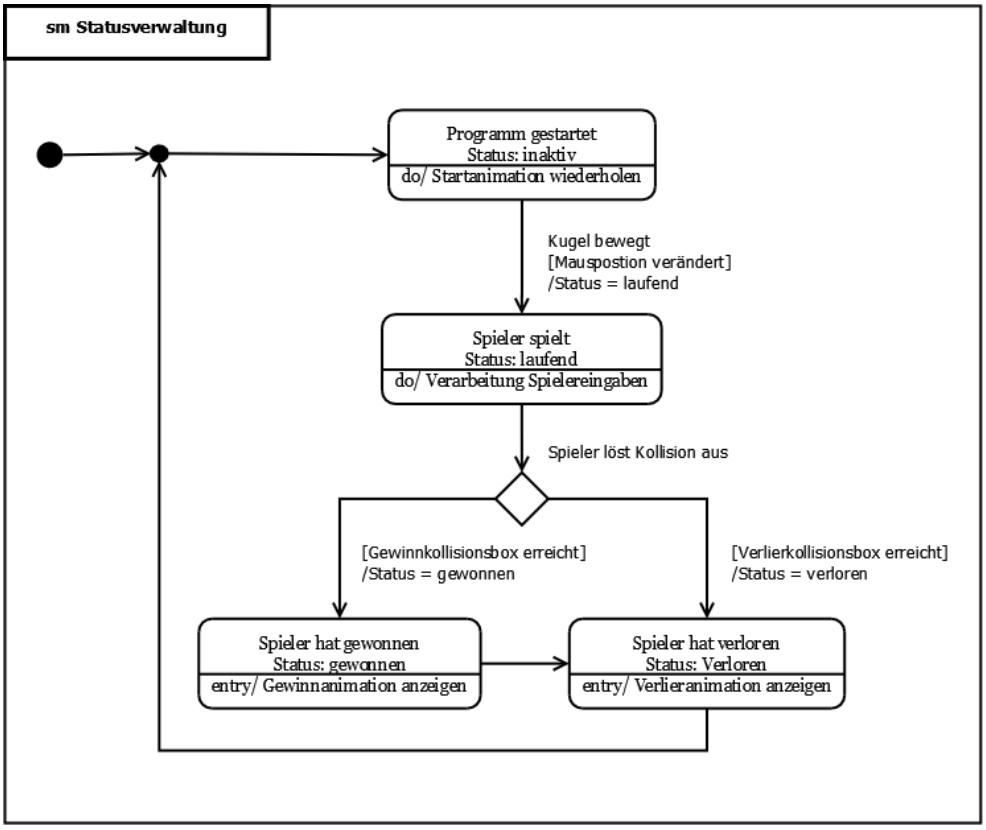

Abbildung A.4.: Verwendete Resourcen

Hardware:

- Büroarbeitsplatz mit Standrechner

Software:

- Windows 7 Professional Service Pack 1 - Betriebssystem
- Unity 3D Version 5.3.3f1 - 3D Entwicklungsumgebung
- Visual Studio 2015 - Code Entwicklungsumgebung
- MiKTEX- Distribution des Textsatzsystems TEX
- TEXMaker - LATEXSchreibprogramm
- Dia Version 0.97.2 - Anwendung zum Zeichnen strukturierter Diagramme
- Entwickler - Implementierung der Scripte / Realisierung
- Anwendungsentwickler - Code Begutachtung

Abbildung A.5.: Model View Controller

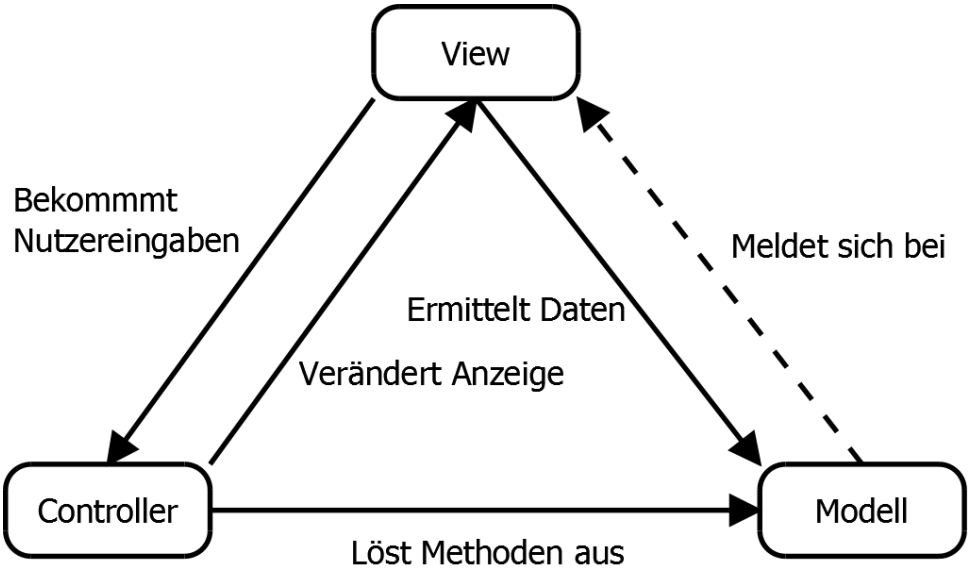

Abbildung A.6.: Presentation Abstraction Control