Meilenstein 3: Benutzerschnittstelle

Abgabetermin: Mittwoch, 27.01.2021, 18:00 Uhr

Aufgabe 1: Schnittstellenarten, Dialoge und Dialogstruktur

Marvelous Mashup ist eine verteilte Anwendung, die aus mehreren Komponenten besteht. Diese Komponenten müssen bestimmte **Schnittstellen** zueinander haben, um interagieren und ihre Systemaufgaben erfüllen zu können. Dazu gehören auch die Schnittstellen in Richtung menschliche Benutzer. Als Akteuere sind mehrere Arten von Benutzern am System beteiligt: mitspielende Benutzer, Zuschauer, Designer von Konfigurationsdateien, Serverbetreiber, KI-Entwickler, usw.

Jeder dieser Benutzer interagiert entsprechend seiner Anwendungsfälle mit bestimmten Komponenten. Um die Bedienung des Spiels so angenehm und effizient wie möglich zu gestalten, muss für jede Komponente eine geeignete **Schnittstellenart** festgelegt werden. Es muss also entschieden werden, ob dieser Teil des Systems über eine Kommandozeile oder eine graphische Oberfläche bedient werden soll. Im Fall von graphischen Oberflächen müssen **Dialoge** definiert werden. Das sind meist eigenständige Fenster oder Anzeigeboxen, die zusammengehörende Informationen darstellen, und über die Benutzer durch Bedienelemente entsprechende Eingaben machen können.

Mit **Dialogstrukturdiagrammen** kann man spezifizieren, welche verschiedenen Dialoge und Dialogwechselmöglichkeiten (also durch welche Aktion man von einem Dialog zu einem anderen übergeht) es in einem System gibt. Dies sind keine UML2-Diagramme, sondern eher informelle Darstellungen, die der Übersicht dienen sollen. Abb. 1 zeigt ein Beispiel.

- Wählt eine **Schnittstellenart** für jede Komponente des Spiels und jeden Anwendungsfall, der Benutzerinteraktionen umfasst. Begründet die getroffenen Entscheidungen, d.h. beschreibt besondere Features der Schnittstelle oder Anforderungen an sie, die die Wahl einer Kommandozeile oder GUI nahelegen.
- Geht von den Anforderungen an das System bzw. von den Anwendungsfällen aus, und gruppiert die vom System zu unterstützenden Handlungen der Akteure zu sinnvollen Einheiten. Jedem Anwendungsfall, der Interaktionen mit einem Benutzer beinhaltet, müssen benannte Dialoge zugeordnet sein, die diese Interaktionen ermöglichen. Dabei können mehrere elementare Interaktionen in einem Dialog zusammengefasst, oder komplexe in mehrere Dialoge unterteilt werden. Listet diese **Dialoge** auf, und gebt dazu an, wie die Anwendungsfälle abgedeckt werden.
- Erstellt für alle Komponenten ein **Dialogstrukturdiagramm**, das die Dialoge und deren Dialogwechselmöglichkeiten zeigt. Denkt auch an kleinere Dialoge, wie Bestätigungsabfragen, Hinweisfenster, Fehlerbenachrichtigungen, usw.

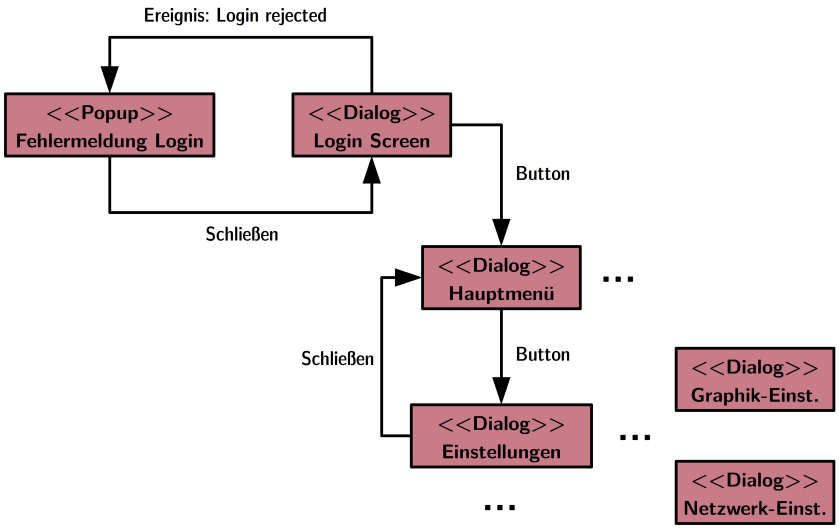

Abbildung 1: Beispiel für ein Dialogstrukturdiagramm.

Aufgabe 2: Graphische Gestaltung und Nutzungskonzept

Nachdem die Dialoge und ihre Übergänge festgelegt sind, muss nun deren graphische Gestaltung konkretisiert werden. An dieser Stelle werden aber noch keine Dialogfenster tatsächlich implementiert, nicht einmal als Prototypen ohne Anwendungslogik. Stattdessen fertigt man **graphische Prototypen** als Zeichnungen an. Für jeden Dialog wird also ein mehr oder weniger abstrakter Entwurf in Form einer **Mock-up-Zeichnung** gemacht. Dazu kann man Diagramm-Tools verwenden, die Paletten mit den üblichen GUI-Widgets haben, oder man zeichnet diese in vereinfachter Form selbst. Hier genügt es, wenn man die Art der GUI-Elemente und ihre Anordnung erkennen kann. Ästhetik oder der look-and-feel einer bestimmten Plattform ist an dieser Stelle noch nicht relevant. Mock-ups sollten simpel und übersichtlich gehalten werden, und einen Eindruck von der wesentlichen strukturellen Gestaltung einer graphischen Oberfläche vermitteln. Damit Betrachter der Mock-ups eine Vorstellung davon bekommen, wie die Oberfläche bedient wird, um bestimmte Interaktionen durchzuführen, werden die graphischen Prototypen von einer **textuellen Beschreibung** der damit möglichen Interaktionen begleitet. Das erfolgt überlichweise im Stil einer Art Benutzeranleitung. Es sollten nicht nur die Eingaben des Benutzers dokumentiert werden, sondern auch die Ausgaben oder dadurch ausgelösten Aktivitäten des Systems. Zusammen stellen die Zeichnungen und Beschreibungen des Bedienungsablaufes ein **Nutzungskonzept** der Schnittstelle dar.

- Erstellt für jeden Dialog einen **graphischen Prototypen,** der die Elemente der GUI und ihre Anordnung zeigt. Für Dialoge, die in gleicher oder sehr ähnlicher Form öfter vorkommen (z.B. Bestätigungsdialoge oder Fehler-Popups) genügt eine einzelne Zeichnung, ggf. mit einer Liste, wo der Dialog auftritt. Verkünstelt euch bei diesen Mock-up-Zeichnungen nicht. Diese dürfen durchaus stilistisch einfach gehalten sein. Prototypen zeigen nur den strukturellen Aufbau der Oberfläche (Art und grobe Anordnung von Elementen), also etwa dass es ein Eingabe-Textfeld mit einem zugehörigen OK-Button darunter gibt. Ob das Feld abgerundete Ecken hat, der Button hübsch violett ist, oder das Längenverhältnis dem Goldenen Schnitt entspricht etc., ist zu diesem Zeitpunkt noch nicht relevant.
- **Beschreibt die Interaktionen,** die über die Dialoge vorgenommen werden können im Stil einer Benutzeranleitung. Ihr könnt dabei eher abstrakt bleiben ("Der Benutzer trägt die Websocket-Adresse des Servers ins Adressfeld ein, und startet eine Verbindung zu diesem mit Klick auf den Connect-Button."). Ihr müsst nicht jede elementare Aktion beschreiben (Mauscursor auf Textfeld bewegen, Klicken für fokussieren, Text eingeben, Maus bewegen auf Connect-Button, Button klicken, ...). Bedienungsmuster, bei denen man voraussetzen kann, dass ein durchschnittlicher Benutzer sie kennt, etwa wie man auf und ab scrollt, müssen nicht in allen Details erklärt werden.