

02 | Choose Your Own Adventure: Der Live-Stream

Aufgabenstellung

Implementieren Sie eine interaktive Geschichte, die an das Konzept der Gamebooks angelehnt ist. Aufgeteilt auf mehrere *Seiten*, werden den SpielerInnen Abschnitte einer zusammenhängenden Geschichte präsentiert. In jedem Abschnitt können die SpielerInnen aus zwei möglichen Optionen für die Fortsetzung der Geschichte wählen und damit die Handlung beeinflussen. In diesem Beispiel erzählen wir die Geschichte des Live-Streams zu diesem Kurs und die Konsequenzen, die das Zuschauen haben kann ;)

Spielablauf

Der Spielablauf kann anhand dieser Grafik nachvollzogen werden. Die entsprechenden Texte finden Sie am Ende dieser Aufgabenbeschreibung.

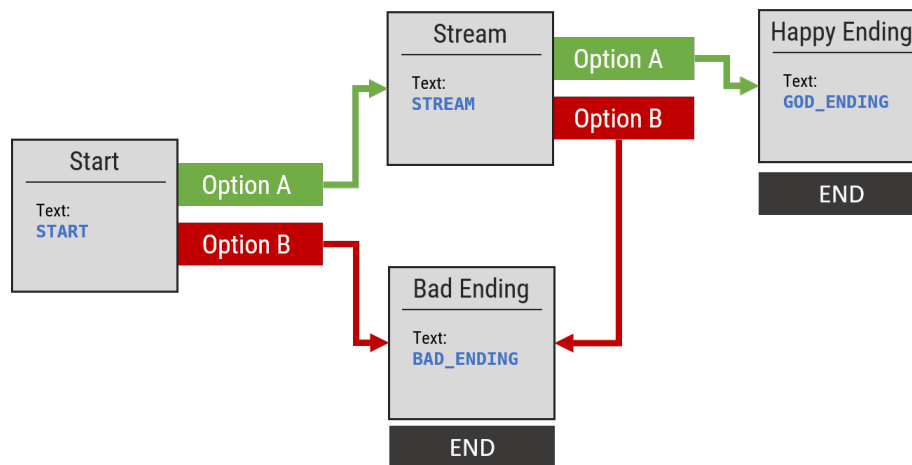


Figure 1: Szenen-Graph der Geschichte

Jede *Szene* besteht aus einem Text und zwei möglichen Optionen für das weitere Vorgehen. Jede Szene wird durch eine *Activity* repräsentiert. Alle *Activities* nutzen das gleiche Layout, in dem *TextView*-Elemente für das Anzeigen von Überschrift und aktuellem *Kapitel* und *Button*-Elemente für die Auswahl der beiden möglichen Optionen verwendet werden.

Vorgehen

Ein neues Projekt

Erstellen Sie in *Android Studio* ein neues Projekt. Wählen Sie als *Project Template* die Option *Empty Activity* und vergeben Sie sinnvolle Werte für den Namen und Paketnamen der Anwendung.

Ressourcen

Für die Anwendung benötigen wir eine Reihe von Texten (*Strings*), die wir an verschiedenen Stellen des *User Interface* verwenden werden. Dazu gehören die einzelnen *Kapitel*, die Beschriftungen für die Buttons zur Auswahl der möglichen Optionen und eine Überschrift, die in jeder *Activity* über dem aktuellen Text angezeigt wird. Grundsätzlich ist es eine gute Idee, Code und Inhalt (hier die Texte) in einer Anwendung bestmöglich zu trennen. In Android nutzen wir dazu Ressourcendateien, in denen wir in diesem Fall Textbausteine speichern um diese im Anschluss im Code unserer Anwendung referenzieren zu können. Inhaltliche Änderungen müssen dann nur noch in dieser XML-Datei vorgenommen werden - die Referenz im Code bleiben unberührt.

Erstellen Sie in der Datei `res\values\strings.xml` Einträge für alle Texte. Verwenden Sie geeignete Bezeichner (Attribut `name`) mit denen die Texte später im Code referenziert werden können. Als Vorlage können Sie das XML-Element mit dem Namen `app_name` verwenden, das automatisch in der Datei angelegt wird.

Layout

In Ihrem Projekt finden Sie bereits eine automatisch erstellte Layout-Datei, die auch bereits in der ersten *Activity* (Methode `setContentView` in der `MainActivity`) verwendet wird. Sie können dieses Layout über die XML-Ansicht oder den graphischen Editor anpassen und die UI-Elemente ergänzen, die Sie zum Umsetzen der Anwendung benötigen: Ein `TextView` für die Überschrift, ein weiteres `TextView` für den Kapiteltext und die beiden `Buttons` zur Auswahl der möglichen Optionen.

Editieren Sie die Layout-Datei `res\layout\activity_main.xml`. Fügen Sie die notwendigen UI-Elemente hinzu und vergeben Sie passende *IDs*, mit denen die Elemente später im Code referenziert und manipuliert werden können.

Die erste Activity

Jede mögliche Szene des Spiels wird durch eine *Activity* repräsentiert. Diese sind dabei sehr ähnlich aufgebaut. Alle *Activities* nutzen das gleiche Layout (`activity_main.xml`) und haben die gleiche Aufgabe:

- Anzeigen des aktuellen Texts

- Anbieten zweier Buttons zur Auswahl der möglichen Verläufe
- Abfangen der Klicks auf diesen Buttons und Wechsel zur damit ausgewählten, nächsten *Activity*

Wir implementieren die vier Szenen des Spiels (**Start**, **Stream**, **Happy Ending** und **Bad Ending**) als individuelle *Activities* um deren Aufbau und den Wechsel zwischen den *Activities* möglichst häufig zu trainieren. Wahrscheinlich wird Ihnen bei der Umsetzung der Aufgabe aber auffallen, dass es für den konkreten Anwendungsfall auch andere, bessere Alternativen zu diesem Vorgehen geben könnte.

Passen Sie die vorhandenen *Activity* an, um hier die erste Szene des Spiels darzustellen. Referenzieren Sie die verschiedenen Elemente des *User Interface* in entsprechenden Variablen. Setzen Sie den korrekten Text aus der Ressourcen-Datei ein. Fangen Sie die Klicks auf die Buttons ab, in dem Sie auf diesen `OnClickListener` registrieren. Nutzen Sie später die *Callback*-Methoden der *Listener*, um über *Intents* zur jeweils ausgewählten, nächsten Szene zu wechseln.

Die anderen Activities

Verwenden Sie angepasste *Activity* als Vorlage für die anderen Szenen, die Sie ebenfalls als einzelne *Activity* abbilden. Neue *Activities* erstellen Sie, in dem Sie zusätzliche Java-Klassen erzeugen, die von der Superklasse `Activity` erben. Denken Sie daran, diese neuen *Activities* im *Manifest* der Anwendung einzutragen. Vergessen Sie dies, wird Ihre Anwendung abstürzen, sobald Sie versuchen zu diesen *Activities* zu wechseln.

Wenn alle Szenen erstellt wurden, können Sie die *Callback*-Methoden der *Listener*, die Sie auf den Buttons registriert haben, verwenden, um über *Intents* zur jeweils passenden nächsten *Activity* zu wechseln. Wählen die SpielerInnen in der ersten Szene z.B. die *Option A* aus, wechseln Sie per *Intent* zur Szene **Stream.**

Erweiterungen

Sobald die grundlegende Funktionalität der Anwendung implementiert ist, können Sie sich an die Verbesserung und Erweiterung machen. Hier finden Sie dazu ein paar Vorschläge:

- Optimieren Sie Ihren Code (*Refactoring*). Überlegen Sie sich, ob Sie die gemeinsame Funktionalität der erstellten *Activities* in eine gemeinsame Superklasse verschieben können, von der die vier *Activities* erben.
- Optimieren Sie das *User Interface* der Anwendung, in dem Sie Textgrößen und Abstände anpassen. Verwenden Sie dazu geeignete XML-Attribute der jeweiligen UI-Elemente. Eine Liste der Gestaltungsmöglichkeiten für z.B. das `TextView` finden Sie hier.

- Seien Sie kreativ: Verändern Sie die Geschichte durch weitere Verzweigungen oder ersetzen Sie sie komplett durch Ihr eigenes Abenteuer.

Texte für die einzelnen Szenen

Start

Freitag. 9 Uhr. Viel zu früh. Das letzte Bier beim Zoom-Stammtisch hätte es gestern Abend eigentlich nicht mehr gebraucht. Während deine Kopf gemächlich seinen Dienst aufnimmt und aus den zwei Schränken vor deinen Augen endlich wieder einer wird, beginnt es in deinem Gehirn zu brodeln. Freitag. 9 Uhr. War da nicht was? Irgendwas mit Robotern und Telefonen? Plötzlich fällt es dir ein! Der Live-Stream zum Android-Kurs beginnt doch jeden Augenblick! Keine Zeit für langes Nachdenken, jetzt muss eine Entscheidung her. Was möchtest du tun?

Option A: Schnell den Laptop holen. Stift und Papier. Ich bereit und will etwas lernen.

Option B: Das Video kann ich mir auch noch später anschauen. Noch 5 Minuten die Augen schließen ...

Stream

Grade noch rechtzeitig schaffst du es, den Stream aufzurufen. Während du der monotonen Stimme des Dozenten lauscht, werden deine Augen wieder schwerer. So ganz kannst du dich immer noch nicht konzentrieren. Und überhaupt. Wer braucht schon Android. Und hier auf Twitch gib es bestimmt auch noch was spannenderes zu schauen. Zusammenreißen und aufpassen oder lieber chillen? Was möchtest du tun?

Option A: Ich reiße mich zusammen. Vielleicht ist das ja doch wichtig und hilft mir irgendwann mal.

Option B: Also zu den Let's Plays von Gronkh kann ich immer ganz gut einschlafen. Schnell den Kanal wechseln.

Happy Ending

Den Stream schaust du dir komplett an. Und das Übungsblatt. Und den Foliensatz gleich auch noch mal. Heute bist du motiviert. Das einzige was dich stört ist, dass du immer noch nicht diese eine App gefunden hast, die du schon ewig suchst. Aber hey, du kennst dich mit Android aus! Hast immer aufgepasst im Kurs. Du setzt dich an den Rechner und beginnst zu programmieren.

6 Monate später. Schon 2 Millionen Menschen haben deine App gekauft. Du schwimmst in Geld. Gut das du dich damals doch aus dem Bett gequält hast. Und gut das du den Live-Stream nie verpasst hast.

Bad Ending

Glücklich schließt du die Augen und bist wenige Sekunden später eingeschlafen. Aber irgendwas stimmt nicht. Du wachst in einem dunklen Raum auf. Das einzige Licht strahlt dir aus einer Deckenlampe, direkt über dir, in die Augen. Ist das ein Traum? Wenn ja, dann wahrscheinlich ein ApItraum. Denn du hörst plötzlich hinter dir langsame, mechanische Schritte und ein schrilles Kichern. Das riesige, grüne Gesicht des Android-Maskottchen schiebt sich in dein Blickfeld. Die großen, regungslosen Augen starren dich an. Der Roboter kichert ohne den Mund zu bewegen.

Das nächste Mal, denkst du dir, bin ich pünktlich zum Live-Stream

Screenshots der Anwendung

