

Beschreibung Architektur Future Bazooka Safari

Wir haben uns entschieden mit dem Framework JavaFX unsere App zu entwickeln. Dazu nutzen wir die IDE NetBeans. Da JavaFX nicht nativ auf Android und IOS läuft, nutzen wir das Plugin von Gluon, welches mit Hilfe von JavaFXPorts und dem AndroidSDK eine Apk für Android erzeugt. Zur Versionierung nutzen wir Github (unter <https://github.com/unitw/Mobile-Praktikum.git>).

Wie man im Entwurf sehen kann enthält unser Projekt das Spiel mit seiner Spiellogik und eine Datenbankkomponente (Shop aus Zeitgründen nicht mehr implementiert). Für das Anmelden und Registrieren nutzen wir JDBC mit dem jTDS JDBC Driver für SQL-Server, da der SQL-JDBC Treiber von Microsoft Probleme auf den Mobilien Geräten machte.

Bevor man sich nun das Projekt über Github klonet, welches NetBeans voll unterstützt, muss man sich als erstes das Gluon Plugin aus dem Plugincenter runterladen und installieren. Nachdem die Installation erfolgreich war, kann man das Projekt öffnen und einsehen. Für das Kompilieren und erstellen einer Apk ist es außerdem wichtig in der Build.gradle, welches sich im Projekt unter Build Scripts/Project finden lässt, den Pfad zu seinem AndroidSDK zu setzen, da es sonst wie oben gesagt nicht funktioniert. Wenn man nun unter dem Knoten Source Packages[Java] guckt sieht man die Packagestruktur.

Im Package *com.futurebazookasafariandroid* befindet sich die Hauptklasse und der *FXMLLoginScreenController*. In JavaFX kann man mithilfe eines GUI-Builders, von Oracle entwickelt und SceneBuilder genannt, Oberflächen designen und einbinden. Diese Scenebuilder erzeugt sogenannte FXML-Dateien. In diesen FXML-Dateien sind alle Informationen über das Layout, Größe, Position hinterlegt. Um Events und andere Aktionen aus der FXML zu empfangen und auszuführen benötigt man die entsprechenden Controller. Dieser Controller wird ebenfalls in der FXML-Datei hinterlegt und verwaltet diese dann. So ist der *FXMLLoginScreenController* der Controller für das *FXMLLoginScreen.fxml*. Das *FXMLLoginScreen.fxml* findet sich unter *Resources[Main]/com.futurebazookasafariandroid*. So liegen alle FXML-Dateien und Bilder im Ordner *Resources[Main]* in einem Package, dass genauso heißt wie das Package unter *Source Package[Java]*.

Im Package *com.futurebazookasafariandroid.Backend* befindet sich die gesamte Logik für das Tower Defense Spiel. Der MapController verwaltet das bewegen der Monster, bauen der Türme, Projektilbewegung und Berechnung des Schadens. Der RundenController verwaltet die Runden, das ein Spieler eine Runde schaffen und nicht schaffen kann. Wenn er es nicht schafft wird alles auf 0 zurückgesetzt und er muss von vorne anfangen. Schafft er es jedoch wird der Rundenzähler erhöht und es wird schwieriger. Der Knoten ist eine Klasse für die Berechnung unseres A-Stern Algorithmus, der die Router der Monster festlegt.

Im Package *com.futurebazookasafariandroid.FBS_DatenBank* befindet sich die Logik für das Einloggen und Registrieren in unserer Datenbank.

Die beiden Packages *com.futurebazookasafariandroid.FBS_HTML* und *com.futurebazookasafariandroid.FBS_HTML.img* sind noch Anfänge des Ingame-Shops der am Ende nicht implementiert wurde.

In *com.futurebazookasafariandroid.FBS_Interfaces* findet man alle Interfaces wieder, die benutzt worden sind. So haben wir für Monster, Level, Hindernisse, Maps, Projektile, Spieler und Tower Interfaces erstellt, da es unter Umständen mehrere verschiedene Typen dieser Komponenten gibt, sie jedoch alle grundlegende gleiche Eigenschaften besitzen.

Das Package *com.futurebazookasafariandroid.FBS_Level* beinhaltet alle Level, die bereits erstellt wurden. Analog dazu die Maps, Monster, Projektile und Tower.

In *com.futurebazookasafariandroid.Frontend* befinden sich der Controller für die Spieloberfläche, welcher die Drag 'n Drop Actions für das Bauen von Türmen und alle Knöpfe verwaltet. Da wir unsere Oberfläche selber zeichnen haben wir die von JavaFX bereitgestellte Komponente *Canvas* erweitert um auf dem Canvas Türme, Monster usw darstellen zu können.