Entwurfsbericht

Synchronisation und Auswertung von Roboterfußballvideos

tj18b

Mitglieder
Dan Häßler
Robert Wagner
Sirk Petzold
Alexander Eichhorn
Erik Diener
Jonas Wrubel
Tomas Daetz Chacon

Betreuer
Hans-Gert Gräbe
Tobias Wieprich
Tobias Jagla
André Köhler

Inhaltsverzeichnis

1 Visionen und Ziele

/LV10/ Das System soll für das NAO-Team HTWK die Bearbeitung und Auswertung von Videos einzelner Roboterfußballspiele vereinfachen und diverse Bearbeitungsmöglichkeiten bereitstellen.

/LZ10/ Das System soll bis zu vier Videos mit festem Dateiformat und festgelegter FPS-Zahl und die dazugehörigen Logs automatisch synchronisieren. Dabei sollen möglichst viele Schritte editierbar und transparent gehalten werden.

2 Rahmenbedingungen und Produktübersicht

Die Applikation gliedert sich aktuell in 5 wesentliche Bereiche.
Als zentrale Einheit dient die **Video-Area**, in der alle geladenen Videos abgespielt und alle Operationen abgebildet werden.
Die **Kontrolleinheit** enthält den Play-Button zum Abspielen der Videos, eine Stop und Skip-Funktion für 1 Frame, sowie eine Skip-Funktion für eine ausgewählte Anzahl von Sekunden. Darüber hinaus gibt es einen Loadbutton für jedes Video, mit dem das geladene Video verändert werden kann. Eine Drag&Drop Funktion soll zusätzlich noch eingeführt werden. Darüber hinaus folgt noch eine Fortschrittsanzeige, mit der sowohl Videos als auch die Logs entsprechend vor- und zurückgespult werden können.
In der **Track-Area** werden später die einzelnen Spuren für die Videos dargestellt, in denen entsprechende Markierungen gesetzt werden können die in der Speicherung im MLT-Format berücksichtigt werden.
Die **Log-Area** dient der Darstellung der geparsesn Logfiles und selektiert stets das Element das zur aktuellen Videoposition passt. Darüber hinaus kann hier per Klick ein Element ausgewählt werden, wodurch alle Videos zu der entsprechenden Stelle gespult werden.
In der **Menüleiste** werden zudem einige zusätzliche Funktionen implementiert. Das Laden von Logfiles, Videos und Bilddateien wird hier ebenso möglich sein, wie das Aufrufen verschiedener Hilfsdokumente, das Auswählen der gewünschten Audiospur und das Rückgängig machen der letzten Aktion.

Die Anwendung ist in ihrer Größe dynamisch veränderlich, läuft aktuell unter Windows und Linux und wird im Laufe der Entwicklung noch einige Verbesserungen bezüglich der Benutzeroberfläche erhalten. Die Darstellung der Videos wird in einem Anwenderfreundlicheren Kontext erweitert. Abdockbare Fenster sind hierbei aktuell im Gespräch.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

3.1 Programmiersprachen/Markup

Die ausschließlich verwendete Programmiersprache ist Java. FXML wird als markup language für das UI verwendet.

3.2 Architektur/Struktur

Die Software kommt ohne eine Server-Anwendung aus und ist eigenständig als Stand-Alone Application funktional.
Als Build-Tool wird Maven verwendet.

3.2.1 Frameworks/Libraries

Die grafische Benutzeroberfläche wird mit JavaFX und dem internen Tool SceneBuilder modelliert. Da aber JavaFX im Bereich Audio- und Videomanipulation einigen unserer Anforderungen nicht gerecht wird, binden wir das Java Framework vlcj ein.

3.3 Vorgehensweise

Zuerst wurde ein Grundgerüst erstellt, welches bereits alle bisher für notwenig erachteten Klassen enthält und in dem der Großteil der geplanten Methoden deklariert ist (siehe Abschnitt „Datenmodell“). Die einzelnen Funktionalitäten werden nacheinander - wie im Releaseplan beschrieben - entwickelt und zusammengeführt.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

4.1 Package: resources

In diesem Package liegen die hierarchisch untergeordneten Packages fxml und styles.

4.1.1 fxml

Hier liegen alle FXML-Dateien, die die hierarchische Struktur und das Layout des GUI beschreiben. Dabei besitzt jede größere Komponente eine eigene fxml-Datei. Das macht den Code übersichtlicher und bringt dadurch Vorteile in der Wartbarkeit und Wiederverwendbarkeit.

4.1.2 styles

In diesem Package liegt die CSS Datei für das GUI.

4.2 Package: java

In diesem Package liegen die hierarchisch untergeordneten Packages Config, GUI_Controller und robosoccer.

4.2.1 Config

Hier befinden sich alle wesentlichen konfigurierbaren Parameter, die das Programm zur Laufzeit benötigt, z.B. die FPS der Videos.

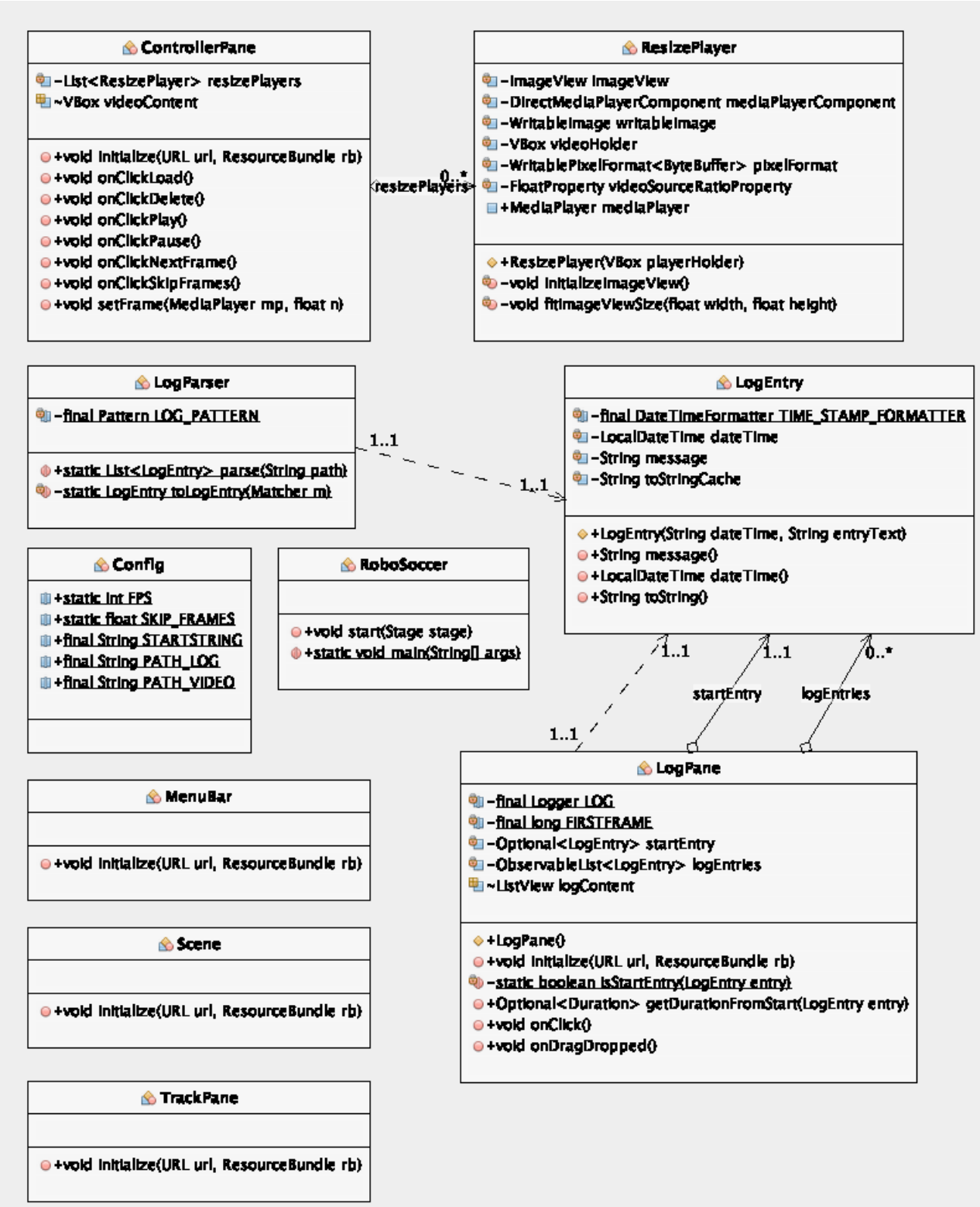
4.2.2 GUI_Controller

Hier liegen alle java-Dateien die unseren GUI Elementen Funktionalitäten verleihen. Die Logik, die hinter den interaktiven Elementen steckt, wird implementiert. Die GUI Controller werden in der FXML angebunden und daher auch über das FXML geladen.

4.2.3 robosoccer

Das Package umfasst die java-Dateien LogEntry, LogParser, ResizePlayer und Robosoccer, welche weitere Logik bezüglich der Bearbeitung der Logs und des Players beinhalten.

5 Datenmodell



6 Glossar

Markup Language (dt.: Auszeichnungssprache) Mithilfe dieser Kategorie von Programmiersprachen werden Inhalte, Formatierungen und Datenaustausch beschrieben, oder weitere Auszeichnungssprachen definiert.

Stand-Alone Application Ist eine Software-Anwendung, welche auf jedem Client-System (Desktop Computer) installiert werden kann und ohne Abhängigkeit zu anderen Services lauffähig ist.

JavaFX Von Oracle entwickeltes Framework, zur professionellen Erstellung plattformübergreifender Java-Applikationen, mit interaktiven und multimedialen GUIs.