Universität Leipzig - Softwaretechnik Praktikum 2014/2015
Entwurfsbeschreibung zum Projekt: Ein kartenbasiertes "Multiplayer"-Spiel

Gruppe: SWT15-GKP

18. Mai 2015

Inhaltsverzeichnis

1 Allgemeines

Im Rahmen des SWT-Praktikums wird ein kartenbasiertes Multiplayerspiel entwickelt, das es ermöglicht, ein an das alte Pacman angelehnte Computerspiel auf einem realen Kartenausschnitt zu spielen. Die entstehenden Highscores werden dann zu den jeweiligen Karten gespeichert um sich mit anderen Spielern messen zu können.

2 Produktübersicht

Der Nutzer kann online über die Spiele-Website auf das Programm zugreifen. Dabei stehen dem Nutzer ein Suchfeld zum finden der gewünschten Spielumgebung zur Verfügung, es kann auch manuell durch Scrollen über die Karte ein Spielort ausgewählt werden, wobei nur Städte ab einer bestimmten Größe zugelassen sind, damit auf jeden Fall eine spielbare Karte erstellt werden kann und die Highscore vergelichbar bleibt. Ebenfalls ist es möglich Spielorte über die Highscoreliste auswählen zu können. Desweiteren besteht die Möglichkeit die Spielfigur zu resetten und die Lautstärke über einen Regler anzupassen. Hat man den Ort ausgewählt, wird das Spielfeld erzeugt und man kann ein neues Spiel beginnen. Man kann den Pucman nun mit Hilfe der Pfeiltasten steuern. Ziel des Spiels ist es den Pucman über die Karte zu steuern und die Kekse die auf der Karte verteilt liegen zu fressen. An dieser Aufgabe wollen den Spieler Geister hindern, die sich auch über das Spielfeld bewegen und Pucman fressen können. Frisst ein Geist den Spieler so wird dieser auf den Startpunkt zurückgesetzt und die Anzahl seiner Leben wird um eins verringert. Es gibt unterschiedliche Geister, die durch ihr aussehen unterschieden werden können und sich anhand verschiedener Logiken bewegen. Falls keine Leben mehr übrig sind, wird das Spiel beendet und die erreichte Highscore zusammen mit der Karten URI und dem Namen des Spielers abgespeichert. Während des Spielens werden die Möglichkeiten zur Veränderung der Kartenansicht deaktiviert. Das Spielgeschehen ist durch einen Soundtrack unterlegt, der eine funktionale und inhaltliche Verbindung zwischen Bild und Ton generiert. Desweiteren werden für das Spiel wichtige Informationen wie aktueller Highscore, die verbleibende Anzahl an Leben und die Möglichkeit direkt auf die Website des Spiels zugreifen zu können angezeigt.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

Als grundsätzliche Programmiersprache haben wir uns für Java-Script entschieden. Wobei der Java-Scriptcode von einer HTML-Seite aufgerufen wird. Desweiteren benutzen wir Phaser als Gameframework. Ein Teil der Software wurde mithilfe der Java-Script Bibliothek JQuery geschrieben, da es keine adäquaten Funktionen in Java-Script gibt. Die von uns umgesetzte Web-Anwendung basiert auf einer Client-Server Architektur. Wir arbeiten also mit einem HTML-Server auf dem die Daten liegen, die vom Client abgefragt und an diesen übertragen werden. Diese Daten beinhalten auch den Gameblock, der dann vom Client ausgeführt wird. Außerdem gibt es noch ein Map-Module, welches das Level aus den Geo-Daten erstellt, die mithilfe der Linked Geo Data API von OpenStreetMaps bezogen werden. Dazu kommt das Highscore-Module, welches für die Verwaltung der Highscores verantwortlich ist. Desweiteren kommt noch ein Datenserver hinzu, der die Highscores unter Verwendung eines Tripplestores speichert.

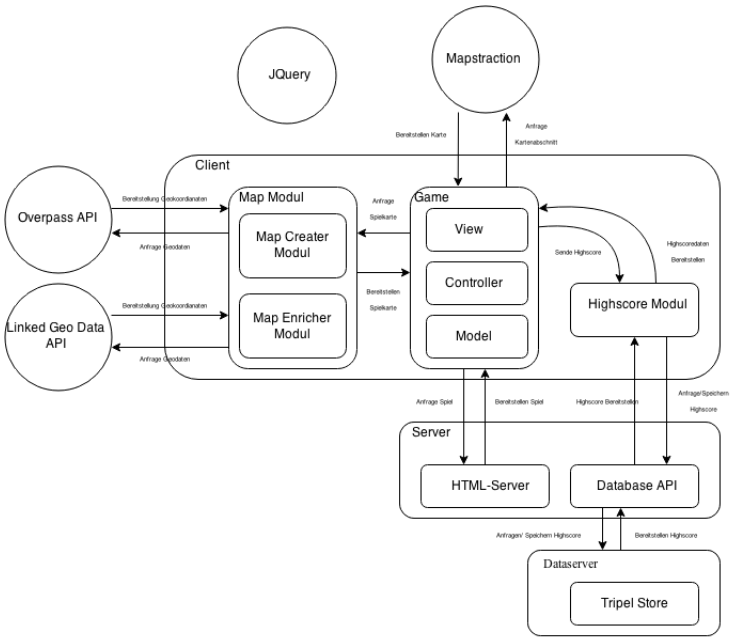

Abbildung 1: Architekur

4 Struktur- und Entwurfsprinzipien der einzelnen Pakete

4.1 Server

Vom Server werden die benötigten Spieldateien zur Verfügung gestellt und automatisch geladen. Desweiteren werden Highscoreanfragen und Manipulationen zwischen dem Dataserver und dem Client verarbeitet und anschließend weitergeleitet.

4.2 Client

Der Gameblock wird an das Model-View-Controller Prinzip angelehnt, welches durch die Nutzung von Phaser umgesetzt wird.

Das Modell (model) enthält die darzustellenden Daten und ist von Präsentation und Steuerung unabhängig.

Die Präsentationsschicht (view) ist für die Darstellung der benötigten Daten aus dem Modell und die Entgegennahme von Benutzerinteraktionen zuständig.

Die Steuerung (controller) verwaltet die Präsentation, nimmt von ihr Benutzeraktionen entgegen, wertet diese aus und agiert entsprechend.

Das Highscore Module verwaltet die Highscores der Spieler, d.h. es stellt Anfragen an den Server zum Triplestore und kann sie über den Server auch abfragen, falls sie im Spiel angezeigt werden sollen.

Die Linked GeoData API extrahiert aus OpenStreetMap die Geo-Daten für das Erstellen der Hintergrundkarte in Abhänigkeit vom gewählten Ort.

Die Overpass API extrahiert die Geo-Daten aus OpenStreetMap, die zum Hervorheben der benutzbaren Straßenzüge und zur Erstellung des Levels benutzt werden.

Das Map Module besteht zum Einen aus dem Map Creator Module, welches lediglich den Graphen des Levels aus den Straßenzügen bildet, und zum Anderen aus dem Map Enricher Module, welches semantische Daten in die Kartenerstellung mit einbindet (z.B. Krankenhäuser, etc.). Das Map Module arbeitet mit den Geo-Daten die von der GeoData API und der Overpass API geliefert werden. Innerhalb dieses Moduls kommen auch Funktionen der JQuery Bibliothek zum Einsatz.

5 Datenmodell

Siehe Abbildung 1.

Beschreibung zu Abbildung 1:
$A \xrightarrow{\text{Datenfluß}} B$ beschreibt den Datenfluss von A nach B, wobei A und B Module der Software sind.

6 Testkonzept

Am Ende des Sprints wird die aktuelle Version der Software auf den Webserver geladen. Diese wird anhand eines Test-Workflows geprüft. Dabei werden die angegebenen Eingaben getätigt und die Reaktion der Software mit der erwarteten Ausgabe verglichen. Sollten dabei Abweichungen auftreten, gilt der Test als nicht bestanden.

7 Glossar

Das Glossar wurde überarbeitet und als externes Dokument angefügt.