

Projekt Web-Programmierung

Leveleditor für 2D Platformer Spiel

Onlinestudiengang Medieninformatik
FH Brandenburg
Modul Web-Programmierung

Projekt von
Martin Freudenberg

Inhaltsverzeichnis

Aufgabenstellung.....	3
Projektbeschreibung.....	3
Umsetzung.....	4
Verzeichnisse.....	8
Dateien.....	8
Architektur.....	9
Quellen der Grafiken.....	10

Aufgabenstellung

Aufgabe war es eine Web 2.0 Anwendung zu erstellen. Dabei sollte das Projekt vor allem folgende Eigenschaften besitzen

"Thematisch gibt es für das Projekt keine Vorgaben; es sollte dabei jedoch folgende Eigenschaften aufweisen:

- Verwendung von AJAX
- Verwendung von CSS, DHTML und DOM
- Integration RSS-Feeds
- Mashups
- keine Tabellen zur Formatierung"

Projektbeschreibung

Der Leveleditor sollte Spieler in die Lage versetzen ein Level für das 2D Platformer Spiel welches ich im Modul Objektorientierte Skriptsprachen erstellt hatte zu erstellen. Dabei ist es möglich den Start- und Endpunkt des Levels zu setzen, unbewegliche und bewegliche Plattformen mit frei definierbarer Farbe zu platzieren, einsammelbare Punkte zu setzen und Zonen zu definieren in denen der Spieler stirbt sollte er hineinlaufen. Die Interaktion mit dem Leveleditor ist über abfragen der Mausposition und der Mausbuttons umgesetzt. Dies ermöglicht eine bequeme Bedienung des Leveleditors auch für unerfahrene Benutzer

Dabei kann der Benutzer mit drei verschiedenen Tools arbeiten. Er kann Objekte platzieren dabei bestimmt er die Position und die Größe des Objektes. Er kann bereits platzierte Objekte auswählen und dessen Parameter ändern und er kann Objekte wieder entfernen.

Sollte der Benutzer fertig sein mit erstellen des Levels so kann er einen Namen für sein Level vergeben und dieses vom Server erstellen lassen woraufhin er einen Downloadbutton angezeigt bekommt. Er kann das Level anschließend downloaden und einfach in den Ordner „levels“ im Spiel schieben, anschließend das Spiel starten und sein Level auswählen zum spielen.

Umsetzung

Allgemein

Der Levleditor ist mit dem HTML5 Tag `<canvas>` umgesetzt wurden. Dabei werden alle platzierten Objekte auf die Canvas gezeichnet. Die Interaktion mit dem Leveleditor ist mit reinem Javascript umgesetzt wurden. Dabei wurde auf objektorientierte Programmieretechnik zurückgegriffen. Der Leveleditor wird alle 40 ms geupdated und alle Objekte die sich gerade im Bereich des Viewports befinden werden neu gezeichnet.

Eingaben

Die Mausabfragen sind mittels Eventhandlern an der Canvas realisiert wurden. Dabei werden globale Variablen mit dem aktuellen Status der Maus gesetzt. (Position, Mousebuttondown, Maus auf Canvas).

Grid

Um das genaue positionieren zu erleichtern wurde das platzieren der Objekte anhand eines Grids (Gitternetz) umgesetzt. Dieses Grid ist 9600x9600 Pixel groß welches auch die maximale Größe für ein Level welches mit dem Leveleditor erstellt werden kann ist. Das Grid wird auf eine nicht sichtbare Canvas gezeichnet um nicht bei jeden Update der Hauptcanvas das Grid komplett neu zeichnen zu müssen. Nun wird immer nur der Teil des Grids auf die Hauptcanvas gezeichnet welcher gerade auf dem Bildschirm zu sehen ist. Gleiches gilt auch für alle anderen Objekte.

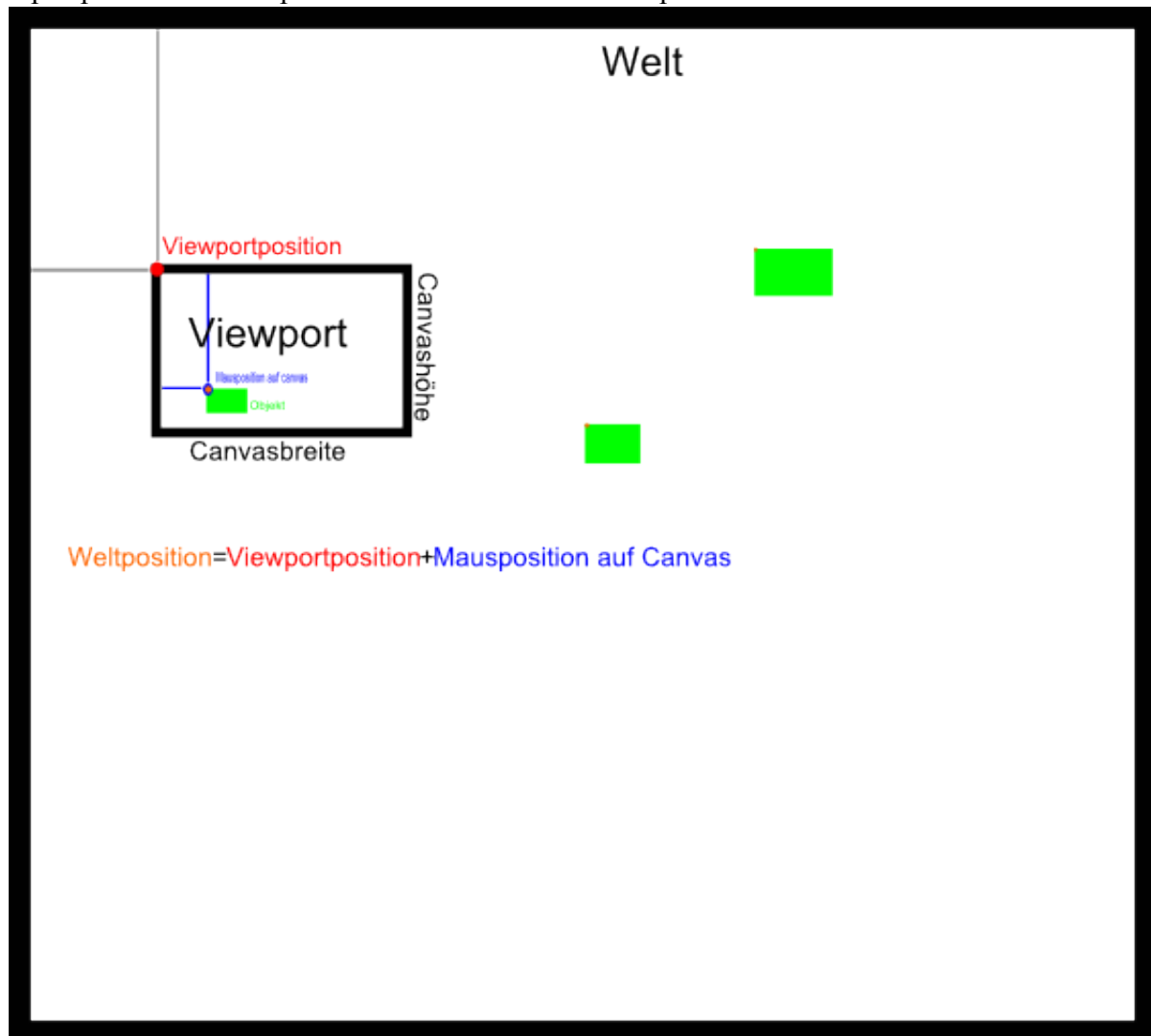
Menü

Das Menü ist mit `<input>` Elementen umgesetzt wurden. Welche beim Event "onclick" die jeweiligen Aktionen im Leveleditor ausführen (wechseln der Tools und Objekt Typen). Bei klicken eines Buttons wird zwischen Tool und Objekttyp umgeschaltet. Dabei werden die gerade aktiven Buttons von Tools und Objekte mittels Javascript mit einer farbigen Border versehen.

Viewport

Ein Level im Editor kann bis zu 9600x9600 Pixel groß sein. Da dieser Bereich zu groß wäre um ihn mit einmal zu Zeichnen wird immer nur der Teil gezeichnet über dem gerade der Viewport ist. Dieser Viewport lässt sich verschieben indem man mit der Maus an den Rand der Canvas fährt. Um später im Spiel genau die Position eines Objektes in der Welt bestimmen zu können müssen die Weltkoordinaten berechnet werden im Leveleditor. Diese ergeben sich aus:

Viewportposition + Mausposition auf der Canvas = Weltposition



Speichern des Levels

Das Level wird vom Server erstellt, dabei werde mittels AJAX die Levelobjektdaten zum Server übertragen. Die Levelobjektdaten befinden sich in einem Javascript Array welches mittels der JSON Funktion stringify() in ein JSON String umgewandelt wird und dann mittels POST übertragen wird. Im PHP Skript wird dann dieser String wieder decodiert mit der Methode json_decode und so ein Mehrdimensionales Array erstellt. Nun wird ein neuer Ordner mit dem aktuellen Zeitstempel + Zufallszahl als Namen erstellt und in diesem eine neue SQLite Datenbank angelegt welche den Namen des vom Benutzer angegebenen Levelnamen hat. Anschließend werden die Levelobjektdaten mittels SQL Querys in die Datenbank eingetragen. Sobald dies geschehen ist wird der Server als Antwort einen Button erstellen mit Verweis auf die eben erstellte Datenbank, dies ermöglicht es dem Benutzer diese herunterzuladen.

Tools

Platzieren:

3 Phasen

1. Maustaste gedrückt → klick Position speichern
2. Maustaste weiterhin gedrückt und bewegt → aktuelle Position – klick Position = Breite und Höhe des neuen Objektes
3. Maustaste losgelassen → aktuelle Position – klick Position = Breite und Höhe des neuen Objektes, Objekt in das Levelobjekte Array eintragen

Verschieben/Auswählen:

3 Phasen

1. Mausbutton gedrückt → schauen ob Maus auf einem Levelobjekt falls ja, Levelobjektposition = Mausposition , Menüeinträge mit Parametern des gewählten Objekt belegen
2. Mausbutton weiterhin gedrückt gehalten → Dragging Levelobjektposition = Mausposition
3. Mausbutton loslassen → Levelobjektposition = Position am Punkt des los lassen der Maus

Entfernen:

Falls Mauspoistion auf Levelobjekt und Maustaste losgelassen → IsDeleted Flag setzen

Daten eines Levelobjekts

X,Y: für die Position des Objektes

Width, Height: Größe des Objektes

Color(r,g,b): Farbe der Plattform

Typ: String der den Typ des Objekts bestimmt

PointTo: Nur relevant bei MovingPlatform Objekt, gibt an wohin sich die Plattform später im Spiel bewegen wird

Velocity: Nur relevant bei MovingPlatform Objekt, gibt an wie schnell sich die Plattform zum PointTo Punkt bewegen wird

Verzeichnisse

buttons: Ordner mit Grafiken für die Buttons des Menüs

img: Bilder die in der .html Datei benutzt werden

levels: Ordner in dem alle Leveldateien erstellt werden mit Unterordner

sprites: Bilder die mit auf die Canvas gezeichnet werden

Dateien

Index.html:

Hauptseite in der der Leveleditor läuft

Leveleditor.js:

LevelEditor Objekt darin definiert, Globale Variablen , main() Funktion zum instanzieren eines LevelEditor Objekts, instanzieren eines Request Objekts und Eventhandler der Canvas hinzufügen

Objekte.js:

Levelentity Objekt darin definiert und weitere Hilfsfunktionen

Request.js:

Objekt welches das Level validiert(Auf bestimmte Kriterien prüft), eine XMLHttpRequest Objekt Instanz anlegt für die Server anfrage und Funktion die bei onreadystatechange ausgeführt wird.

Tools.js:

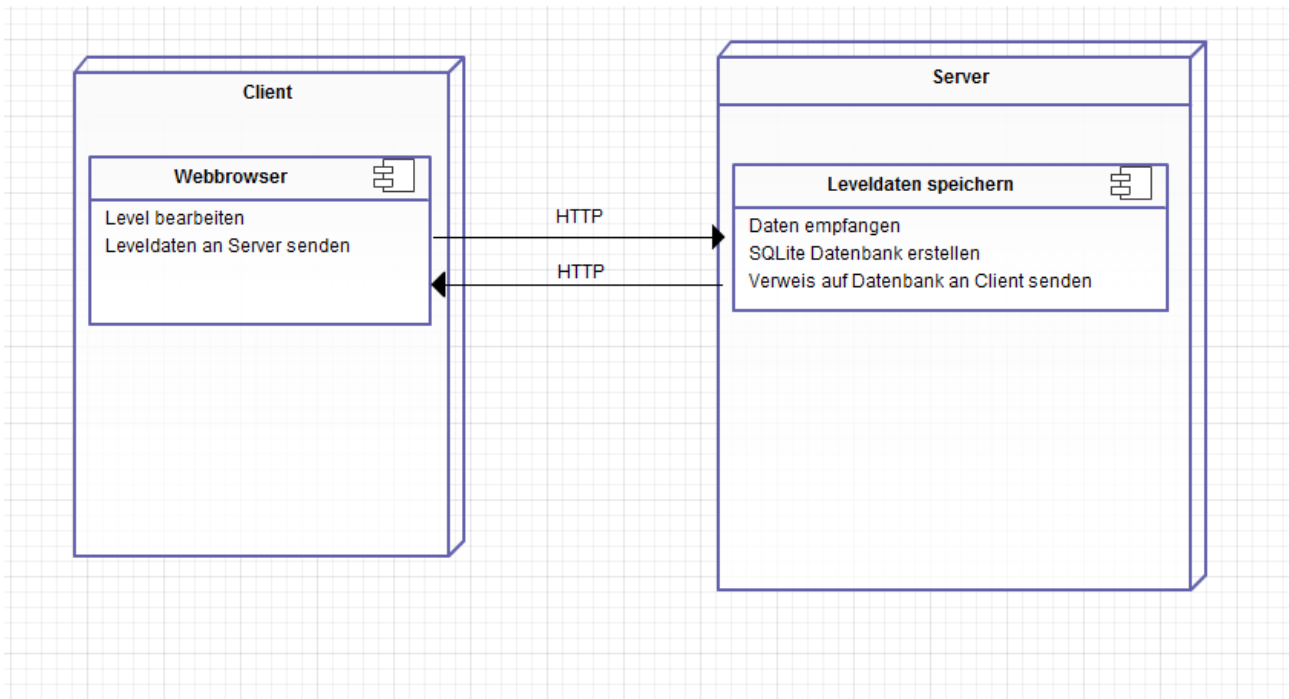
Tools Objekte die im Leveleditor benutzt werden können darin definiert

Savelevel.php:

PHP Skript welches die POST Anfrage auswertet und eine SQLite Datenbank erstellt in der die Levelobjekte gespeichert wurden

style.css: Aussehen der Website definiert darin

Architektur



Quellen der Grafiken

loading.gif generiert auf

<http://www.ajaxload.info/>

„Generated gifs are totally free for use“