Ausarbeitung - PSE - Pascal Jäschke

Projekt

Für das Modul „Praxis im Software Engineering“ sollte ein Spiel namens „Space Taxi“ als Ergebnis erstellt werden. Das Spiel soll unterschiedliche Levels beinhalten, indem eine Spielfigur auf ein Spielfeld von Ort A nach Ort B kommen soll, um in den nächsten Level zu gelangen. Gegnerische Spielobjekte sollen eingebaut werden, die dieses Vorhaben verhindern. Des Weiteren sollen Hindernisse eingebaut werden, die einen Durchgang der Spielobjekte verhindern und quasi als „Wand“ gelten.

Vorgehensweise:

Als ich von der Aufgabenstellung hörte, habe ich mir zunächst im Internet einige Beispiele angesehen und angespielt. Die Funktionsweise war recht simpel und schien sehr leicht umsetzbar. Jedoch habe ich so früh damit begonnen, dass erst später bekannt wurde, dass das Spiel auch in Unity umgesetzt werden kann. Da ich bereits so weite Fortschritte in HTML5 und JavaScript gemacht und von Unity wenig Kenntnisse hatte, entschied ich mich dafür weiter bei meiner HTML5 + JavaScript - Lösung zu bleiben.

Spiel Ausführen:

Aufgerufen werden kann das Spiel über die Seite <http://spacetaxi.zz.mu/SpaceTaxiWar/>.

Login:

Es wurde eine Login-Form implementiert, in dem ein Username ausgewählt werden kann. Diese darf jedoch gewisse Sonderzeichen nicht enthalten. Dafür wird auf einen regulären Ausdruck „[^a-zA-Z0-9\_äöüÄÖÜß]“ geprüft, in dem nur einige der gängigsten Zeichen erlaubt sind.

Tutorial:

Loggt sich ein User ein, dessen Usernamen bisher unbekannt ist, wird ein Tutorial aufgerufen, in der die Basisinformationen und eine kurze Ansprache aufgeführt werden.

Spielfeld:

Es wird ein Spielfeld benötigt, auf dem sich der Spieler bewegt. Dazu kam ein Canvas-Feld sehr gelegen. Ich fand glücklicherweise früh die Bibliothek „KineticJS“, was viele Funktionalitäten bereithielt, wie das Einbinden weiterer Spielfiguren, die Bewegungen, später auch das Einbinden von Sprites. Nun habe ich mir eine Möglichkeit überlegt, wie ich die Stages speichern kann. Dazu kam eine Datenbank in Frage oder ein Filesystem, wofür ich mich letztendlich entschied. Darin war zunächst ein Ordner enthalten, in dem die „default“-Stages gespeichert wurden. Diese bestimmen die Werte (Höhe und Breite) des Spielfeldes, sowie all derer Figuren. Nun mussten diese nur noch ausgewählt werden können. Ich entschied mich dafür ein einfaches DropDown-Feld zu verwenden.

Gestartet werden musste das Spiel natürlich auch. Zu Beginn wurde ein Button am oberen Rand gesetzt, auf den man drücken musste. Das war allerdings unvorteilhaft, da die Figuren mit der Tastatur gesteuert werden sollten. Also wurde die „Enter“-Taste dafür überschrieben. Mit anderen ausgewählten (später änderbaren) Tasten kann die Richtung des Spielers gesteuert werden. Da die normale Steuerung jedoch zu einfach war, wurde eine Gravitation implementiert, die den Spieler, je nach Schwierigkeitsgrad, schneller nach unten zieht. Dementsprechend werden auch mehr Punkte beim Einsammeln eines Passagiers vergeben.

Besondere Schwierigkeiten gab es bei der Überprüfung einer Kollision. Bei Unity war diese Funktionalität bereits gegeben. In JavaScript allerdings müssen pro Frame stets alle Figuren selbst auf eine Kollision miteinander überprüft werden. Um nicht zu viele Überprüfungen pro Frame zu vollziehen, was sich stark auf die Performance auswirkt, wurde die Framerate verringert, sodass weniger Schritte durchlaufen werden müssen. Es gibt folgende Spielfiguren:

Spieler (Player): Den Spieler selbst. Dieser wird, bis auf die Schüsse (Bullet) in jedem Frame auf alle weiteren Figuren geprüft.

Schuss (Bullet): Diese Figur wird von der Position des Spielers aus abgeschossen. Diese prüft auf Kollisionen aller Figuren, bis auf den Spieler selbst, der nicht getroffen werden kann. Ist diese Figur außerhalb des Spielfelds oder trifft sie eine andere Figur, wird sie entfernt. Einen Zusatz gibt es bei den Passagieren, wobei ein „Friendly Fire“ deaktiviert werden kann. In dem Fall wird eine Kollision mit einem Passagier ignoriert.

Wand (Wall): Eine Wand, die im Spielfeld platziert werden kann, um den Spieler zu stören. Eine Kollision mit einer Wand verhindert das Weiterfliegen in diese Richtung.

Passagier (Victim): Es müssen pro Durchlauf alle Passagiere eingesammelt oder bei aktivierten Friendly Fire abgeschossen werden. Wird ein Passagier eingesammelt, erhöht sich der Punktestand abhängig vom eingestellten Schwierigkeitsgrad. Wird einer abgeschossen, werden die Punkte deutlich verringert.

Gegner/Alien (Enemy): Kollidiert der Spieler mit einem Gegner ist das Spiel vorbei und gilt als verloren. Trifft ein Schuss einen Gegner, so werden beide Objekte (Gegner und Schuss) entfernt und der Punktestand erhöht sich um 5.

Ziel: Kollidiert der Spieler mit dem Ziel, so wird geprüft, ob sich noch Passagiere auf dem Spielfeld befinden. Sind noch welche vorhanden, wird eine entsprechende Meldung ausgegeben und das Spiel als verloren beendet. Sind keine Passagiere mehr vorhanden, gilt das Spiel als gewonnen und es wird die nächste Stage (falls vorhanden) geladen.

Ziel:

Ziel des Spiels ist es, pro Durchlauf in kürzester Zeit möglichst viele Passagiere einzusammeln, möglichst viele der gegnerischen Konkurrenz (Aliens) abzuschießen und anschließend zur Basis zu fliegen. Je besser diese Kriterien erfüllt sind, desto mehr Punkte werden verteilt, sodass sich der persönliche Highscore erhöhen kann. Natürlich steht der Spaßfaktor immer an erste Stelle ;)

Optionen:

Ein besonderes Augenmerk gebührt dem Optionsmenü, in dem diverse Anpassungen am Spielverlauf geändert und pro User gespeichert werden können. Folgende Änderungen können dort gespeichert werden:

Sprache: Die Anzeigesprache im Spiel.

Töne: Effekt-Sounds während des Spiels.

Hintergrundmusik: Während der gesamten Zeit auf dieser Seite.

Friendly Fire: (De)Aktiviert das Abschießen der Passagiere. Am Ende einer erfolgreichen Runde wird auf diesen Wert geprüft. Ist Dieses Feld aktiviert, wird ein Bonus mit dem Faktor 3 auf die Gesamtpunktzahl gewährt.

Schwierigkeitsgrad: Bestimmt die Gravitationsgeschwindigkeit des Spielers nach unten. Je höher dieser Wert, desto mehr Punkte werden beim Einsammeln der Passagiere vergeben.

Richtungstasten:

Hoch, Runter, Links, Rechts, Turbo: Ändert die Steuerung mit der Tastatur.

Der Schuss und das Resetten sind mit der Leer- und Entertaste vorbelegt und sind nicht änderbar.

Bilder ändern:

Spieler-Bild, Ziel-Bild, Passagier-Bild, Alien-Bild: Hier können pro User die vordefinierten Bilder ersetzt werden. Kommt ein „Besucher“ im Stage-Builder auf diese Stage, werden auch diese hier definierten Bilder angezeigt.

Anzahl Sprites:

Spieler-Sprites, Ziel-Sprites, Passagier-Sprites, Alien-Sprites: Es besteht die Möglichkeit pro Bild Sprites zu hinterlegen. Bei einem Wert von 1 gibt es keine Animationen für dieses Bild und dieses wird auf eine maximale Höhe und Breite von 50 Pixel runterskaliert. Wird nun eine Sprite-Angabe von mehr als 1 angegeben, so wird die Breite des hochgeladenen Bildes durch diese Anzahl geteilt und als Animation im Spiel angezeigt. Wichtig für eine korrekte Darstellung ist, dass alle Sprites in derselben Höhe dargestellt werden können und alle Einzelbilder dieselbe Breite haben.

Es ist eine Größe von mindestens 1 und maximal 100 an der Anzahl der Sprites definiert.

Stage-Builder:

Ein weiteres Highlight ist der Stage-Builder. Im Stage-Builder können eigene Stages erstellt und gespielt werden. Auch andere User können die hier erstellten Stages betreten und spielen. Ein im Stage-Builder erreichter Highscore wird zwar kenntlich gemacht, jedoch nicht gespeichert.

Wurde bei „User-besuchen“ der eigene User angewählt, kann diese nun geändert und gespeichert werden. Um die Höhe und Breite des Spielfeldes zu ändern, wurden zwei DropDown-Menüs eingebunden. Beim Ändern eines dieser Felder vergrößert oder verkleinert sich das Spielfeld sofort. Um eine Figur hinzuzufügen bzw. zu entfernen, muss auf die jeweilige Figur ein Doppelklick ausgeführt werden. Es folgt eine Meldung, in der nun abgefragt wird, ob diese Figur entfernt oder erstellt (dupliziert) werden soll. Es muss jedoch von jeder Figur mindestens eine vorhanden sein. Das Löschen und Hinzufügen weiterer Spieler oder Ziele ist nicht möglich.

Bewegt werden die Figuren einfach per Drag & Drop.

Wurde die Stage nach eigenem Ermessen geändert, muss diese mit einem Klick auf „Speichern“ gespeichert werden. Dabei hat der User wieder zwei Möglichkeiten. Einmal die vorhandene Stage zu überschreiben oder eine neue zu erstellen.

Erst wenn eine neue oder geänderte Stage gespeichert wurde, kann diese gespielt werden. Wird eine gerade zu ändernde Stage gewechselt, gehen alle nicht gespeicherten Änderungen daran verloren.

Zeit:

In jeder Stage geht es auch um Zeit. Je schneller der Spieler am Ziel ist desto mehr Punkte werden ihm zugerechnet. Die Zeit ergibt sich aus der Höhe und Breite des Spielfeldes und kann nicht separat geändert werden.

Bekannte Bugs:

Beim Starten des Spiels ruckelt es gelegentlich. Es ist jedoch kein Bug, sondern das Scrollen des Fensters zum Spieler, der immer im Mittelpunkt stehen muss.

Wird das Spiel einmal „massakriert“ (zig male resettet oder beim Fliegen gegen einen Spieler im richtigen Zeitpunkt resettet), wird die aktuelle Stage unkontrolliert immer wieder neu geladen. Ein neu laden des Fensters (F5) behebt dieses Problem. (Scheinbar behoben (weiter testen))

Gelegentlich können einzelne Bilder, die eine „Übergröße“ besitzen auf dem Spielfeld in Originalgröße angezeigt werden. Sobald alle Bilder vollständig geladen wurden, wird die Stage auch neu aufgebaut und die vergrößerten Bilder erhalten die normale Größe.

Verwendete Komponente:

Es wurde die jquery-Bibliothek Version 2.1.1 verwendet, sowie die KineticJS-Bibliothek.

Zuvor wurde eine WordPress-Seite mit dem Hoster „Hostinger“ erstellt und das Projekt nach Fertigstellung dort integriert. Zusätzlich ist das gesamte Projekt über GitHub zugänglich und kann von dort heruntergeladen und über einen lokalen Host gestartet werden.