*Tom Auer, Khang Ho, Ewald Reinhardt*

Technische Beschreibung

Super Square Blobster

Stand 18.02.2016

Technische Vorrausetzungen

* **Web-Server:** Um volle Funktion des Spiels zu gewährleisten, also um Audio-, JSON- und Bilddateien laden zu können, wird ein Webserver benötigt. Zur Sicherheit wird nicht unter dem „file://“ System gearbeitet. Vielmehr wird das http-file System genutzt (<http://blog.chromium.org/2008/12/security-in-depth-local-web-pages.html>). Dies ist nötig, um die Auslieferung statischen Webinhalts zu ermöglichen.
* **Game Engine:** Für die technische Umsetzung der Features des Spiels, wie etwa von Animationen oder Kameraausrichtung, ist die Verwendung einer Game Engine bzw. einer entsprechenden JavaScript Bibliothek unerlässlich.
* **Persistente Speicherung von Daten**: Für Fortschritte im Spiel oder einen Highscore wird eventuell das persistente Speichern von Daten genutzt.
* **Tools zur Erstellung von Assets**: *Inkscape* oder ähnliche Programme

Technische Recherche

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Framework / Library | Ausgesucht weil | Testing | Mögliche Alternativen |
| Game Engine (Animationen, Audio, Spiel-mechanik, etc.) | Phaser | Bei der Recherche im Internet, der Befragungen von Kommilitonen sowie beim Einarbeiten in das Platformer Tutorial auf der offiziellen Website von Phaser hat sich diese Game Engine als für unsere Zwecke am geeignetsten herausgestellt. So ermöglicht Phaser beispielsweise die effiziente Implementierung von Animationen, Sprites, Sounds sowie der Game-Physik und der Kameraausrichtung. | Platformer Tutorials auf der offiziellen Phaser Website | MelonJS, CraftyJS |
| Web-Server | Node (express) | Node erlaubt die Implementierung der Server-Komponente in JavaScript. Durch den Zugriff auf die zahlreichen Pakete (npm) können alle notwendigen Funktionen umgesetzt werden. Außerdem ist die grundsätzliche Funktionalität bereits aus dem Kurs bekannt. Für das Ausliefern des HTML-Content und das Routing der REST-Schnittstelle wird das aus dem Kurs bekannte express-Modul verwendet. Dies ermöglicht die effiziente Umsetzung der oben beschriebenen Voraussetzung. | Durcharbeit der Beispiele aus dem Kurs. Coding eines DEMO-Projekts: **Statisches Ausliefern eines HTML-Dokuments**, Textuelle Anwort auf einen GET-Request über eine URL-Route, Lokale Installation von Mongodb und Verbindung mit der Datenbank über den Node-Client | Vertx, Apache Tomcat, PHP-Eigenbau, ... |

Links zu den verwendeten Libraries

* Phaser: <http://phaser.io/>
* Node.js: <https://nodejs.org/en/>