Projektskizze Docker

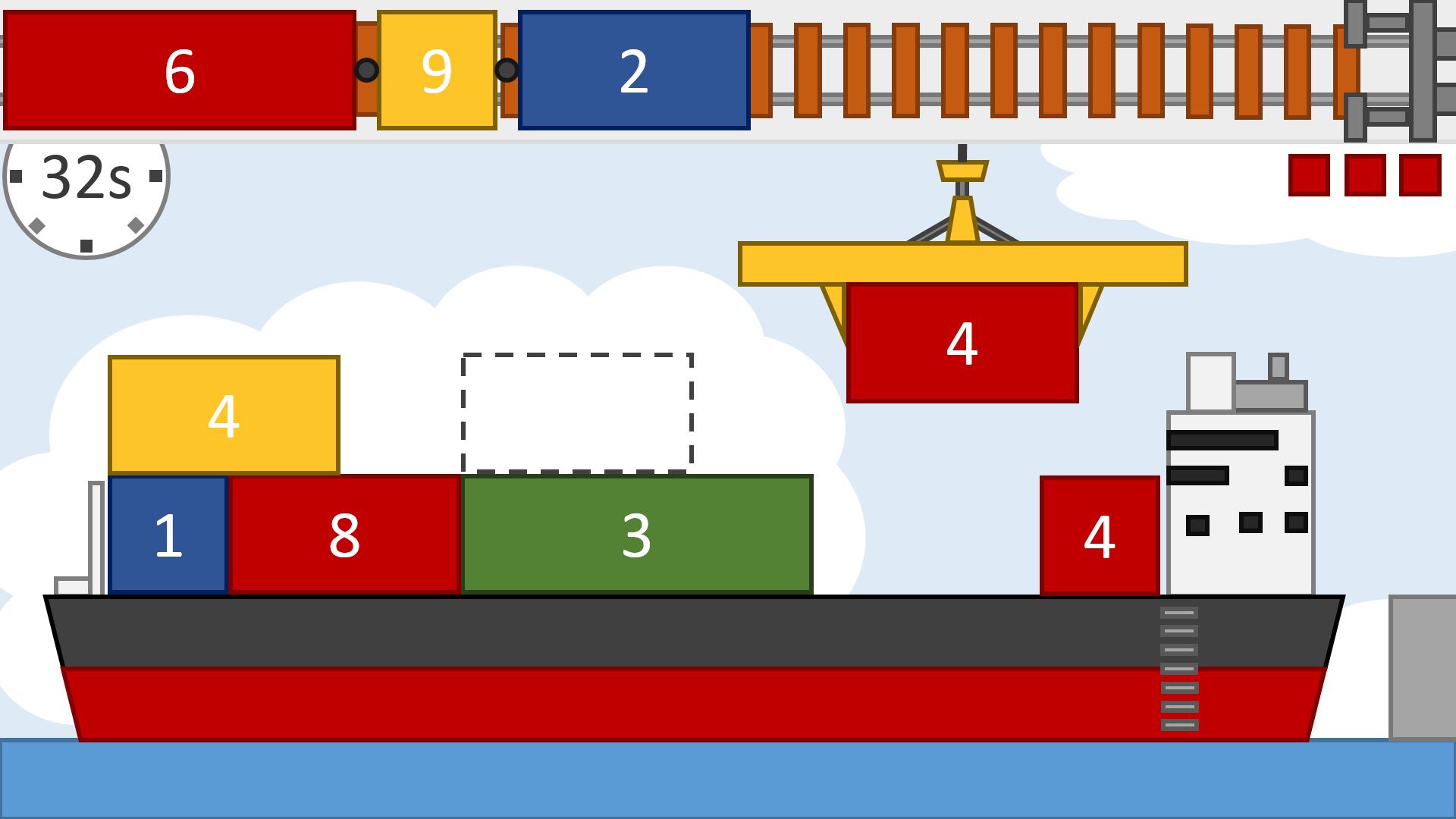
# Idee

Für das Modul SEPS soll innert 14 Wochen ein Softwareprojekt durchgeführt werden, mit dem Fokus auf Projektmanagement und objektorientiertes Design.

Ziemlich schnell war für das Team klar, dass eine Android Applikation entwickelt werden sollte. Die Bedienung und das Konzept sollte demnach auf einen Touchscreen angepasst werden. Die anfängliche eines Rollenspiels wurde jedoch wegen fehlender Innovationen für die Bedienung wieder verworfen. Spontan kamen wir auf die Idee das Ur-Spiel Tetris mit weiteren Themen zu paaren, wobei es zur schlussendlichen Idee für das Spiel Docker kam.

Beim Spiel Docker geht es darum ein Frachtschiff mit Containern zu beladen, wobei diese möglichst kompakt und gleichmässig verteilt verladen werden müssen. Der Spieler muss dabei nicht nur geschickt sein, sondern auch gegen die Zeit ankämpfen.

# Hauptanwendungsfall



## Das Spielziel

Die Aufgabe besteht darin, Container in einer vorgegebenen Zeit, möglichst effizient auf einem Frachtschiff zu verladen. Das Gewicht soll so verteilt werden, dass an keinem Ort eine übermässige Belastung besteht oder allzu hoch gestapelt werden muss.

## Der Spielablauf

Im oberen Bildschirmabschnitt fährt ein Güterzug mit den zu verladenden Containern von links nach rechts. Die Container müssen in der vorgegeben Reihenfolge mit dem Kran auf dem Schiff positioniert werden. Sie haben verschiedene Grössen und Gewichte. Wenn ein Container den rechten Bildschirmrand erreicht ist er verloren.

Das Spiel ist zu Ende wenn:

* das Schiff fertig beladen ist
* mehr als zwei Container verloren gehen
* die Zeit abläuft

## Punkteverteilung

Erst wenn das Spiel zu Ende ist zeigt sich, ob gut gestapelt wurde. Ist die Last an einem Punkt zu zentral zerbricht es, ist sie einseitig sinkt das Schiff und es führt zu einem Punkteabzug. Zusatzpunkte verdient man sich

# Weitere Anforderungen

## Nicht-Funktional

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Anforderung | Beschreibung | Priorität |
| 1 | Entwicklungsplattform Android | Ziel ist, am Ende des Projekts eine lauffähige Android-Applikation zu haben. | Hoch |
| 2 | Systemsprache Englisch | Die Systemsprache der Applikation soll Englisch sein. | Hoch |
| 3 | Schneller Spieleinstieg | Für Mobile-Spiele ist ein schneller Einstieg ins Spielgeschehen wichtig. Dazu soll es möglichst wenige Bedienhürden etc. geben. | Mittel |
| 4 | Kurze Ladezeiten | Hängt mit dem schnellen Spieleinstieg zusammen. Die Ladezeit soll möglichst kurz, aber sicher unter 10 Sekunden gehalten werden. | Mittel |
| 5 | Geringe Speicheranforderung | Mit Rücksicht auf den begrenzten Mobiltelefonspeicher und evtl. langsame Datenverbindungen sollte das Produkt möglichst wenig, aber sicher weniger als 10 Megabyte Speicher beanspruchen. | Mittel |
| 6 | Intuitive Bedienung | Die Bedienung der Anwendung, sowohl die Menüs als auch das Spiel selber, sollen möglichst einfach und intuitiv gehalten werden. | Hoch |
| 7 | Ansprechendes Design | Die Anwendung soll optisch ansprechend und konsistent gestaltet werden. | Mittel |

## Abgrenzung

* Eine iOS-Version wird im Rahmen dieses Projekts nicht angestrebt. Grund dafür ist hauptsächlich die nicht vorhandene Hardware (iPhones, Mac OS X).
* Zusätzliche Desktop- oder HTML5-Versionen sind rein optional.
* Es ist keine Mehrsprachigkeit geplant.

# Ressourcen

Für die Projektrealisation des Android-Spiels Docker wird ein Team von 4 Personen vorausgesetzt. Alle Teammitglieder sollten Erfahrung in der Programmierung mit Java besitzen. Weitere Programmierkenntnisse sowie Spezialwissen in OOP und Softwaredesign sind auf jeden Fall erwünscht.

Fehlendes Wissen in Verbindung mit Android muss auf jeden Fall aufgearbeitet werden.

Da eine grundlegende Spielphysik für die Realisierung notwendig ist, sind ebenfalls Kenntnisse in Physik und Mathematik unabdingbar.

* Remo Höppli (RH)
  + Projektleitung
  + Etwas Programmiererfahrung (ZHAW Niveau)
* Yacine Mekesser (YM)
  + Softwareentwicklung (Python, Java)
  + Etwas Android-Erfahrung
* Emil Wangler (EW)
  + Softwareentwicklung (Java)
  + Etwas Android-Erfahrung
* Christoph Mathis (CM)
  + Softwareentwicklung (Java)
  + Etwas Android-Erfahrung

Der Gesamtaufwand für die Realisierung von Docker wird auf ca. 10 „Mann-Wochen“ geschätzt (400 Arbeitsstunden).

# Risiken

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Risiko** | **Beschreibung** | **EW** | **AW** | **Massnahmen** |
| 1 | ZHAW Netzwerk | ZHAW Server sind aufgrund eines Wartungsfensters oder Ausfalls nicht erreichbar | sehr wahr- scheinlich | Gering | Git benutzen |
| 2 | Motivation | Motivation während des Semesters lässt nach | wahr-scheinlich | Mittel | Arbeiten gerecht verteilen. Teamgeist pflegen und klare gemeinsame Ziele definieren |
| 3 | Probleme mit der Entwicklungsumgebung | Probleme mit Framework oder Android SDK | möglich | Hoch | Gemeinsames Einrichten der Entwicklungsumgebungen und gegenseitige Unterstützung bei Problemen |
| 4 | Hardware | Ein Handy oder Notebook fällt aus | möglich | Hoch | Material sorgfältig behandeln und bei einem Ausfall zeitig für Ersatz sorgen |
| 5 | Sound & Grafik | Zeit für die Implementation wird knapp, Mittel für die Realisierung reichen nicht aus | möglich | Hoch | Sound weglassen und/oder Grafik vereinfachen |
| 6 | Personaldefizit | Ausfälle durch Krankheit oder Unfall, viel zu tun bei der Arbeit | möglich | Mittel | Velo Helm aufsetzen, Rechtsvortritt beachten und jeden Tag ein Glas O'saft trinken. Viel Wissenstransfer |
| 7 | Schlechtes Zeitmanagement | Fehleinschätzung, Zeitmangel auf Grund von Teilzeit Pensum | unwahr- scheinlich | Hoch | Realistischen Zeitplan erstellen. Verzögerungen frühzeitig erkennen und aufholen |
| 8 | Know-how Defizit | Das Know-how im Team oder bei einzelnen Mitgliedern führt zu Verzögerungen | möglich | Gering | So viel Wissenstransfer betreiben wie möglich |
| **EW: Eintrittswahrscheinlichkeit AW: Auswirkung** | | |  |  |  |

## Risikodiagramm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Eintretens Wahrscheinlichkeit →** | sehr wahr- scheinlich | 1 |  |  |
| wahrscheinlich |  | 2 |  |
| möglich | 8 | 6 | 3, 4, 5 |
| unwahrscheinlich |  |  | 7 |
|  |  | Niedrig | Mittel | Hoch |
|  |  | **Auswirkung →** | | |

# Grobplanung

|  |  |  |
| --- | --- | --- |
| **Phase** | **Iteration** | **Ziele** |
| Inception | I1 | Projektskizze erstellt, IDE eingerichtet, erste Ausformulierung der Anwendungsfälle, erster Entwurf der Architektur |
| Elaboration | E1 | Anwendungsfälle definiert, Architektur und Domänenmodell fertig, GUI Designkonzept/Prototyp erstellt, 10% der Programmierungstasks abgeschlossen |
| Construction | C1 | 50% der Programmierungstasks abgeschlossen |
| Construction | C2 | 90% der Programmierungstasks abgeschlossen, GUI Fertig |
| Transition | T1 | 100% der Programmierungstasks abgeschlossen, Testing, Dokumentation und Abschlussarbeiten |
|  |  |  |
| Iterationsdauer: 2 -3 Wochen | |  |

## Zeitplan

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23. Sep | **30. Sep** | 07. Okt | 14. Okt | **21. Okt** | 28. Okt | 04. Nov | 11. Nov | **18. Nov** | 25. Nov | 02. Dez | **09. Dez** |
| **I1** | | **E1** | | | **C1** | | **C2** | | **T1** | | |
|  | **M1** |  |  | **M2** |  |  |  | **M3** |  |  | **M4** |

|  |  |  |
| --- | --- | --- |
| Meilensteine: | 30.09.2014 | Präsentation Projektskizze |
|  | 21.10.2014 | Präsentation Anforderungen |
|  | 18.11.2014 | Präsentationen Design |
|  | 09.12.2014 | Schlusspräsentationen |

# Kundennutzen

Im Vordergrund diese Spiels sollen der Spass und die Kurzweiligkeit stehen. Es soll die Möglichkeit bieten ein kleines Spiel zwischen durch zu Spielen aber auch für längere Zeit unterhaltsam zu sein.  
Der Spieler wird immer mit neunen Herausforderungen konfrontiert, welcher er in einer gewissen Zeit erfüllen muss. Dabei wird auf das logische vorausschauende und kombinatorische Denken speziellen Wert gelegt.

# Wirtschaftlichkeit

Um den Kostenaufwand zu errechnen werden die Geschätzen zu leistenden Stunden zusammen gerechnet. Wir rechnen damit, dass wir pro Woche ungefähr 4 Stunden pro Person für dieses Projekt aufwenden werden und das 12 Wochen lang. Das würde bedeuten, dass in dem Projekt am Ende 192 Projektstunden enthalten sind. Dabei wird ein Stundenansatz von 80 Schweizer Franken gerechnet, womit das Projekt auf einen Gesamtbetrag von 15360.- kommt.

Es ist ein Spiel welches es in dieser Form noch nie erschienen ist, was uns auf dem Markt eine gute Positionierung zulassen wird.

Um diese Kosten zu decken kann das Spiel im Play Store verkauft werden. Bei einem Betrag von 2.- müssten 7680 Kunden unser Produkt kaufen. Wen mit 10 neunen Kunden pro Woche gerechnet wird dauert es somit etwa 15 Jahre bis das Spiel bezahlt ist.

Das Spiel könnte auch gratis abgegeben werden, damit könnte eine grössere und schnellere Verbreitung erreicht werden. Zusätzlich könnte dann mit Werbung im Spiel gewisse Erträge gemacht werden und es könnten verschiedene In-App Verkäufe zur Verfügung gestellt werden, welche es den Spieler das Spiel erleichtern. Durch die grössere Verbreitung ist es eher möglich das Produkt in einer angemessenen Zeit, so 2-3 Jahre zu amortisieren. Die grössere Verbreitung hilft vor allem dabei wen man plant weitere Spiele auf den Markt zu bringen.