Projektskizze Docker

# Idee

emily

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

Ziemlich schnell war für das Team klar, dass eine Android Applikation entwickelt werden sollte. Die Steuerung sollte demnach auf einen Touchscreen angepasst werden. Die erste Idee des Teams bestand darin eine Art Rollenspiel zu programmieren. Ideen für das Spiel bestanden reichlich, jedoch fehlte es an Innovationen für die Bedienung. Spontan kamen wir auf die Idee eine Art 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 muss. Dabei gilt es natürlich ein Zeitlimit einzuhalten.

# Hauptanwendungsfall

emily

 -> yacine

Spielablauf

Der Spieler die Aufgabe, Container möglichst effizient auf einem Frachtschiff zu verladen.

Zu Beginn ist das Schiff noch leer. Im oberen Bildschirmabschnitt fährt ein Güterzug mit den zu verladenden Containern von links nach rechts hindurch. Diese Container gilt es der Reihe nach zu verladen. Container haben unterschiedliche Grössen und Gewichte. Der Spieler steuert dabei den Kran und platziert damit die Container auf dem Frachtschiff. Dabei gilt es, die Last möglichst gleichmässig auf der Ladefläche zu verteilen.

* Ziel
  + Last möglichst gleichmässig auf der Ladefläche zu verteilen.
    - D.h. an keinem ort soll ein übermässige Gewichtsbelastung vorherrschen.
    - Die Container sollen möglichst flach gestapelt werden.
* Ende
  + Schiff ist fertig beladen
  + Container gehen verloren, wenn sie die rechte Bildschirmseite erreichen
  + Mehr als n Container gehen verloren
  + Evtl. zusätzliches Zeitlimit?

Das Spielprinzip ist noch provisorisch

# Weitere Anforderungen

yacine

## Funktional

## Nicht-Funktional

* Kurze Ladezeiten
  + für schnellen spieleinstieg, wichtig für mobile oder so
* Intuitive Bedienung
* Systemsprache Englisch
* Hauptentwicklungsplattform ist Android
* (optional) Schöne Grafik

# Ressourcen

Remo

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 Entwicklungs- umgebung | 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 | möglich | Mittel | Velo Helm aufsetzen, Rechtsvortritt beachten und jeden Tag ein Glas O'saft trinken |
| 7 | Schlechtes Zeitmanagement | Fehleinschätzung, Zeitmangel auf Grund von Teilzeit Pensum | unwahr- scheinlich | Hoch | Realistischer Zeitplan erstellen. Verzögerungen frühzeitig erkennen und aufholen |
| 8 | Knowhow Defizit | Das Knowhow im Team oder bei einzelnen Mitgliedern führt zu Verzögerungen | möglich | Gering | So viel Wissenstransfer betreiben wie möglich |
| **EW: Eintretens Wahrscheinlich AW: Auswirkung** | | |  |  |  |

## Risikodiagramm

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

# Grobplanung

|  |  |  |
| --- | --- | --- |
| **Phase** | **Iteration** | **Ziele** |
| Inception 1 | 1 | Projektskizze erstellt, IDE eingerichtet, erste Ausformulierung der Anwendungsfälle, erster Entwurf der Architektur |
| Elaboration 1 | 2 | Anwendungsfälle definiert, Architektur und Domänenmodell fertig, GUI Designkonzept/Prototyp erstellt, 10% der Programmierungstasks abgeschlossen |
| Construction 1 | 3 | 50% der Programmierungstasks abgeschlossen |
| Construction 2 | 4 | 90% der Programmierungstasks abgeschlossen, GUI Fertig |
| Transition 1 | 5 | 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** |
| **Inception 1** | | **Elaboration 1** | | | **Construction 1** | | **Construction 2** | | **Transition 1** | | |
|  | **M1** |  |  | **M2** |  |  |  | **M3** |  |  | **M4** |

|  |  |  |
| --- | --- | --- |
| Meilensteine: | 23.09.2014 | Präsentation Projektskizze |
|  | 21.10.2014 | Präsentation Anforderungen |
|  | 18.11.2014 | Präsentationen Design |
|  | 09.12.2014 | Schlusspräsentationen |

# Kundennutzen

christoph

* Spass
* Zeitvertreib
* Fördern des logischen, vorausschauenden und kombinatorischen Denkens

# Wirtschaftlichkeit

christoph

Potentieller ertrag durch werbung oder kosten im appstore

Evtl. aber besser gratis und werbefrei releasen, wenn kein grosser gewinn absehbar (lieber verbrelitung)