

Planeten Browsergame

Tim Felix Tanner (Matrikelnr)

Bastian Schneider (1151420)

Einleitung

- Projektbeschreibung
- Projektziel
- Projektbegründung
- Projektschnittstellen
- Projektabgrenzung

Stand der Technik

Anforderungsdokumentation

- Übersicht
- Verlauf
- Erste Konzepte
 - Landscape Layout
 - Portrait Layout
- Finales Konzept

Architekturbeschreibung

Implementierung

- Spielkonzepte

Tests und Usability

- Usability
 - Benutzeroberfläche

Zusammenfassung

- Soll-/Ist-Vergleich
- Lessons learned

Einleitung

Die Folgende Projektbeschreibung schildert den Aufbau und Ablauf des Projekts im Modul Mobile Applikationen der Fachhochschule Bielefeld, welches wir im Laufe des Semesters durchgeführt haben.

Projektbeschreibung

Die Entwicklung eines Spiels, das auf Browserspielen basiert, die früher populär waren. Das Thema des Spiels ist, aufgrund der Tatsache dass der Fantasie kaum Grenzen gesetzt sind, Science-Fiction. Das Spiel findet in einem fiktivem Universum statt und der Spieler übernimmt die Rolle des Herrschers eines Planeten.

Projektziel

Die entwickelte Applikation soll anschaulich und intuitiv sein. Es wird bewusst auf Animationen und ähnliches verzichtet, um den Stil und Charme der vorher genannten Browserspiele beizubehalten. Die Benutzeroberfläche soll simpel gehalten werden und dennoch alle nötigen Funktionen klar erkennbar darstellen um eine einfache Bedienung zu gewährleisten. Das Design soll einheitlich sein um der Applikation ein natürliches Aussehen zu verleihen.

Das Spielkonzept soll einfach gehalten werden und trotzdem dafür sorgen dem Spieler Spaß und die Lust zum weiterzuspielen bringen. Berechnungen und ähnliches sollen daher auch im Hintergrund geschehen um den Fluss des Spiels so wenig wie möglich zu unterbrechen.

Projektbegründung

Aufgrund der Tatsache, dass kaum Browser-Spiele, so wie sie sind, als Mobile Applikationen veröffentlicht wurden, haben wir uns dazu entschieden, dieses Projekt zu bearbeiten.

Projektschnittstellen

Die Applikation, die vom Benutzer bedient wird, ist nur ein Teil eines Systems. Die Eingaben des Benutzers werden an einen Server gesendet und dort verarbeitet. Der Server wurde mithilfe von [Node.js](#) programmiert. Des Weiteren gibt es eine [MySQL](#) Datenbank zur Speicherung von verschiedensten Werten. Es wurde außerdem [MySQL Workbench](#) zur Überwachung und zum Einpflegen der Daten verwendet. Die Kommunikation zwischen Server und Datenbank ist auch mit [Node.js](#) geregelt und im Server implementiert. Genauere Informationen befinden sich im Kapitel [Implementierung](#). Der Benutzer interagiert also nur mit dem Client, während alles andere an anderer Stelle passiert. Mehr dazu im Kapitel, in dem die [Architektur](#) beschrieben wird.

Projektabgrenzung

Da die Arbeitszeit am Projekt beschränkt ist, soll die Applikation mit wenigen Spielern im Sinn implementiert werden.

Stand der Technik

Anforderungsdokumentation

Übersicht

Must have:

- ☒ Basen können ausgebaut werden (Minen, Truppen)
- ☒ Angriffe auf Spieler
- ☒ Ressourcenabbau
- ☒ Bilder
- ☒ Server-Client Architektur
- ☒ Datenbank
- ☒ Mindestens drei verschiedene Einheiten mit mindestens zwei verschiedenen Werten (Angriff, Leben)
- ☒ Funktionierendes Kampfsystem
- ☒ Erstellung eines Kontos und Anmeldung

Should have:

- ☐ Nicht-Spieler-Charaktere
- ☒ Forschung
- ☒ Verbesserungen (Truppen/Gebäude)
- ☒ Verteidigungsanlagen
- ☒ Ansprechenderes Design
- ☒ Einfaches Kampfsystem

Could have:

- ☐ zweidimensionale Karte
- ☐ neue Server (Verschiedene Spielmodi)
- ☐ Sound bzw. Musik
- ☒ Mehr Einheitenauswahl
- ☒ Komplexes, ausbalanciertes Kampfsystem

Won't have:

- ☐ Animationen
- ☐ Werbung
- ☐ Bildschirmdrehung

Verlauf

Zu Beginn des Projekts waren wir sehr motiviert und haben geglaubt, dass wir auch die could have's teilweise noch erreichen könnten. Im Laufe der Arbeit am Projekt hat sich herausgestellt, dass diese Einschätzung sehr optimistisch war und wir haben uns darauf geeinigt, uns mit den should have's zufrieden zu geben. Am Ende ist es dann doch noch dazu gekommen, dass wir unsere erste Zielsetzung erreicht haben.

Erste Konzepte

Landscape Layout

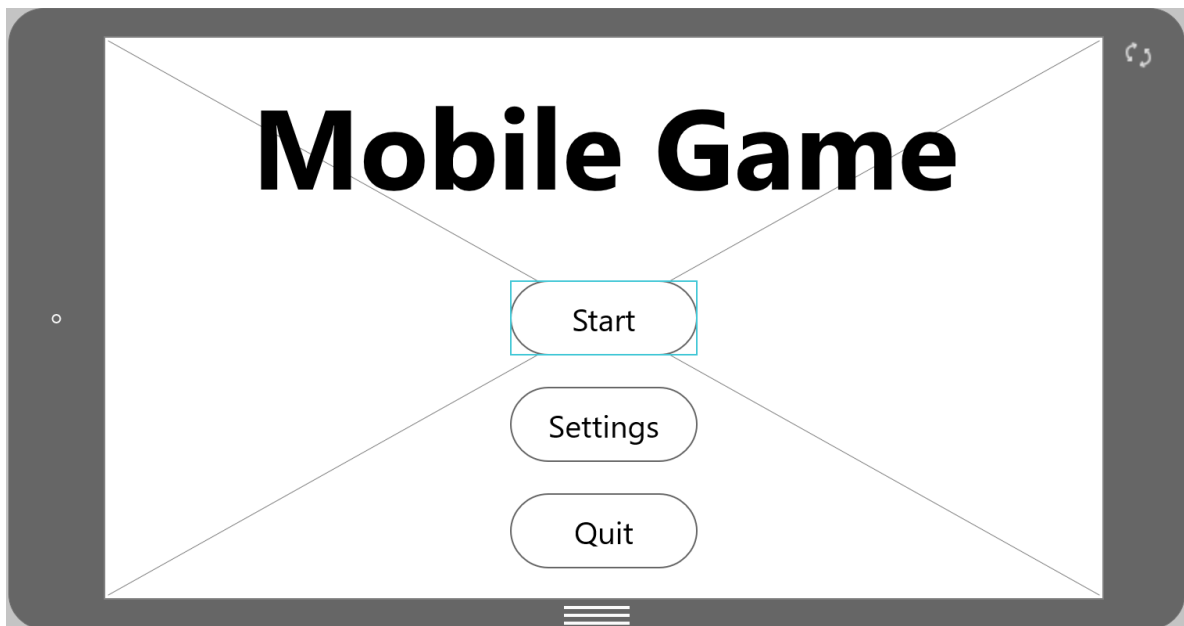


Bild 1: Mockup der Startseite

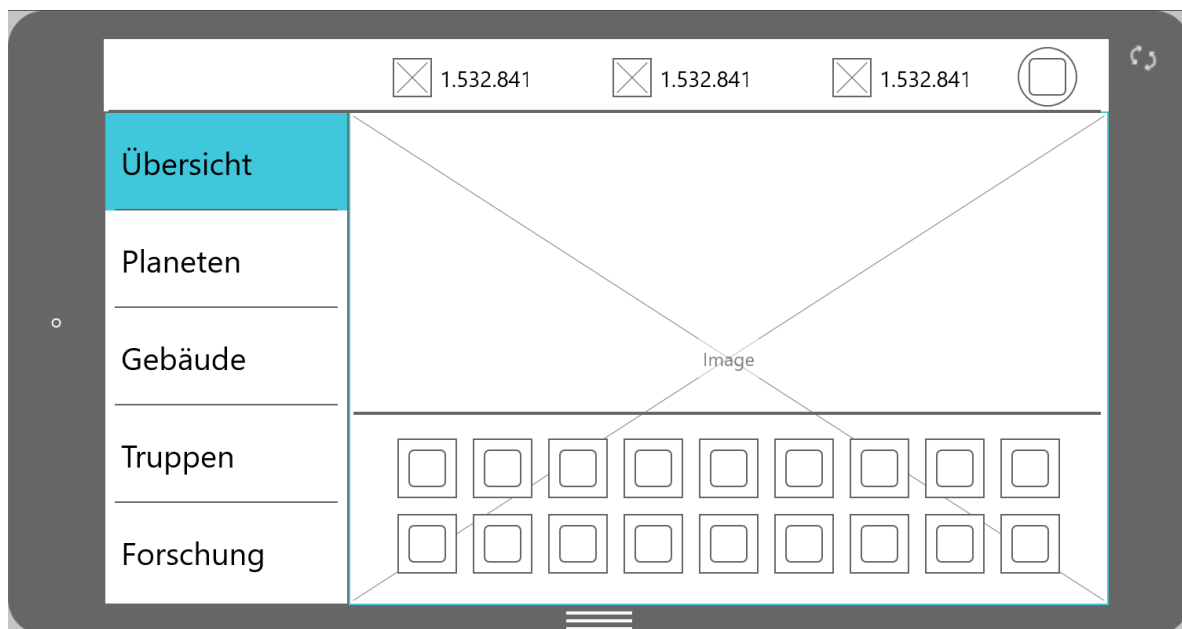


Bild 2: Mockup der Übersichtsseite

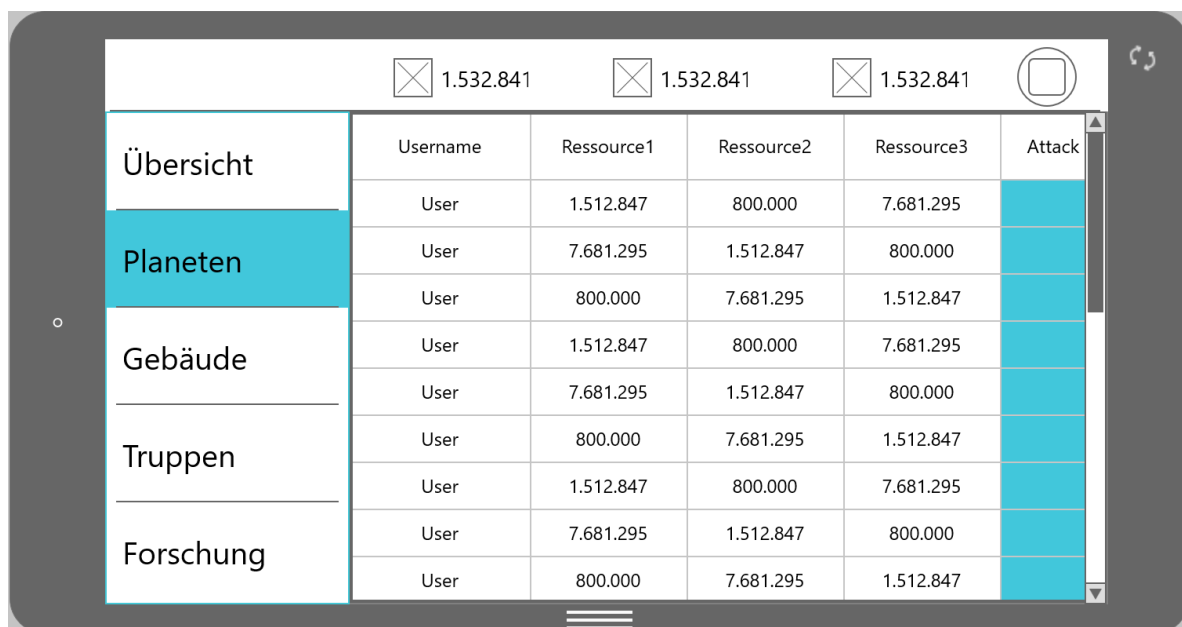


Bild 3: Mockup der Planetenseite

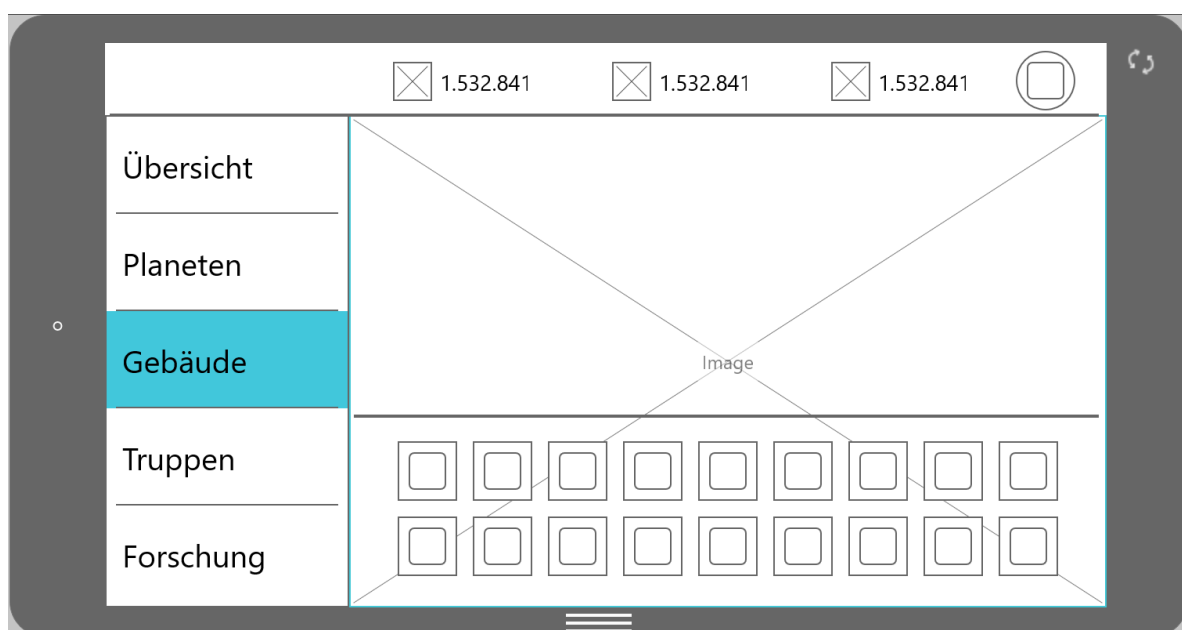


Bild 4: Mockup der Gebäudeseite

Portrait Layout

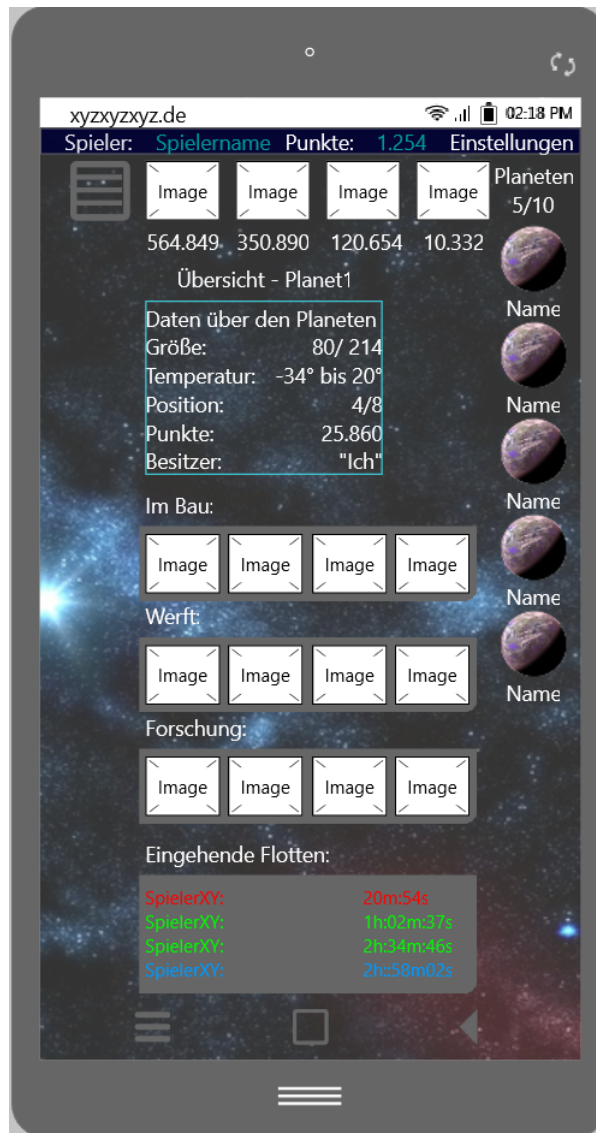


Bild 5: Mockup der Übersichtsseite

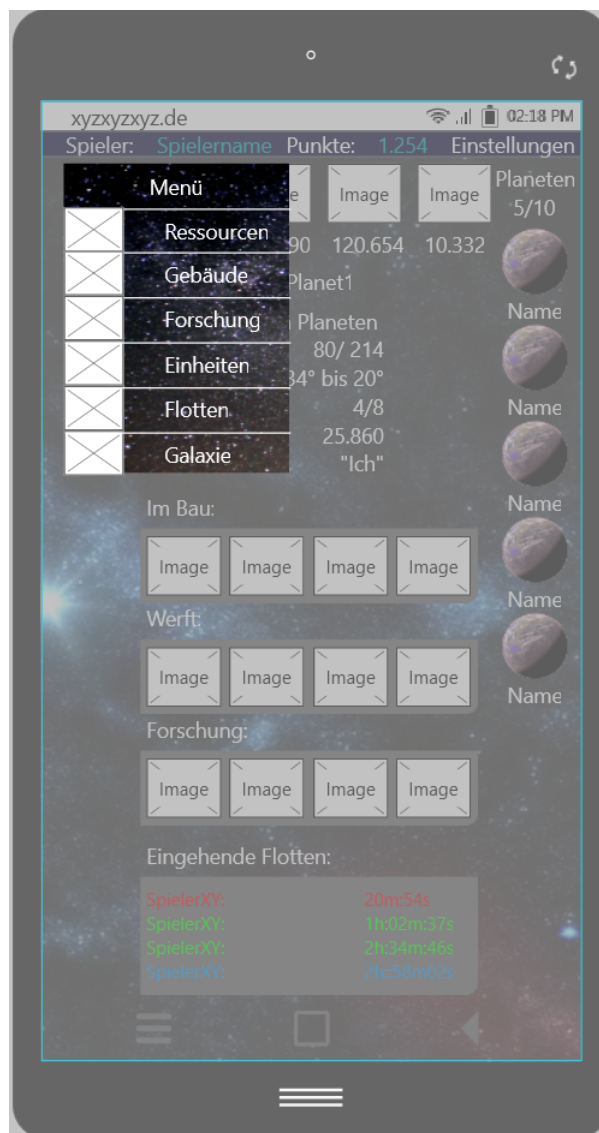


Bild 6: Mockup des Dropdown Menüs

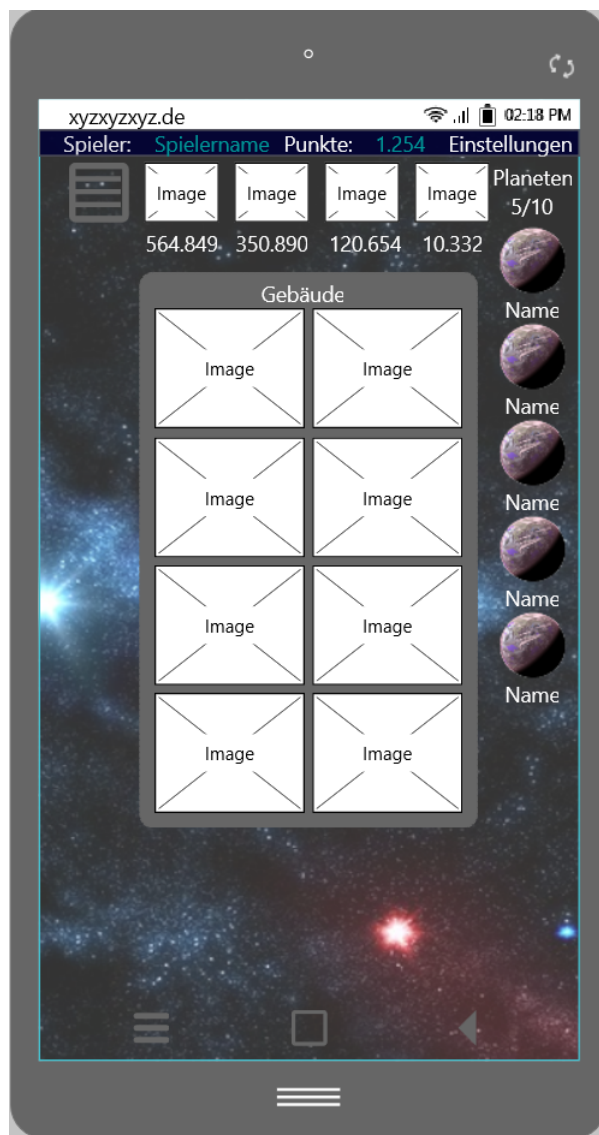


Bild 7: Mockup der Ressourcenseite

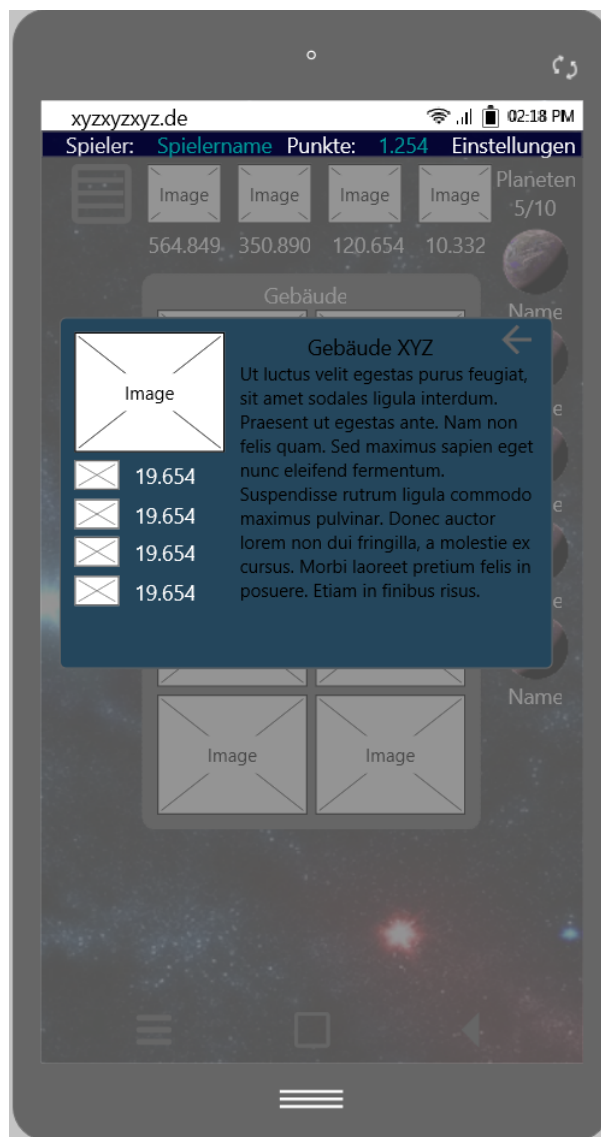


Bild 8: Mockup der Detailansicht

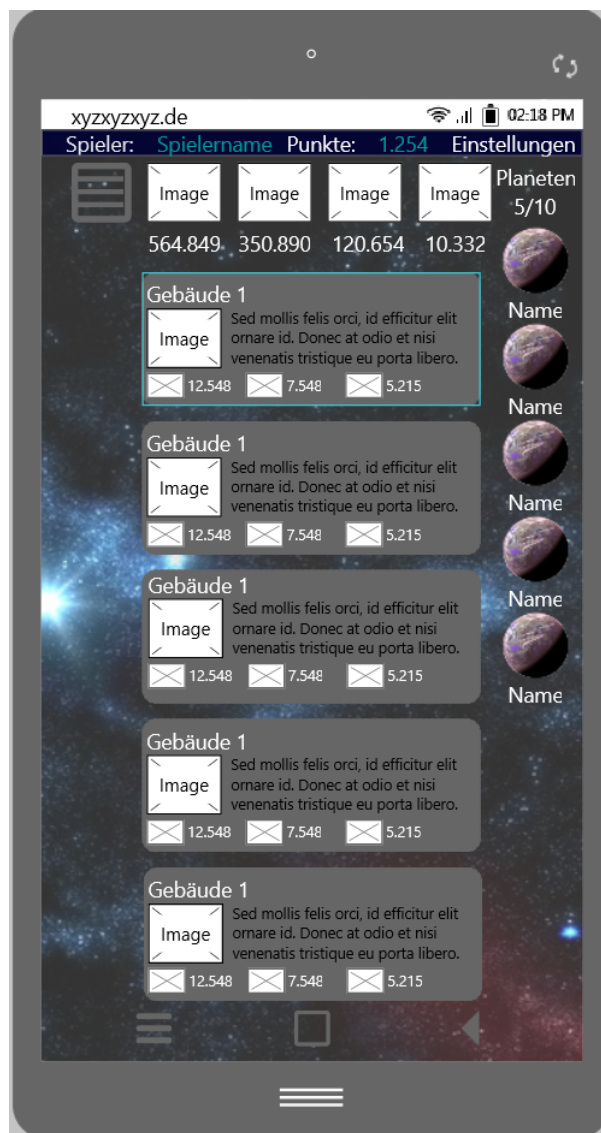


Bild 9: Mockup der Gebäudeseite

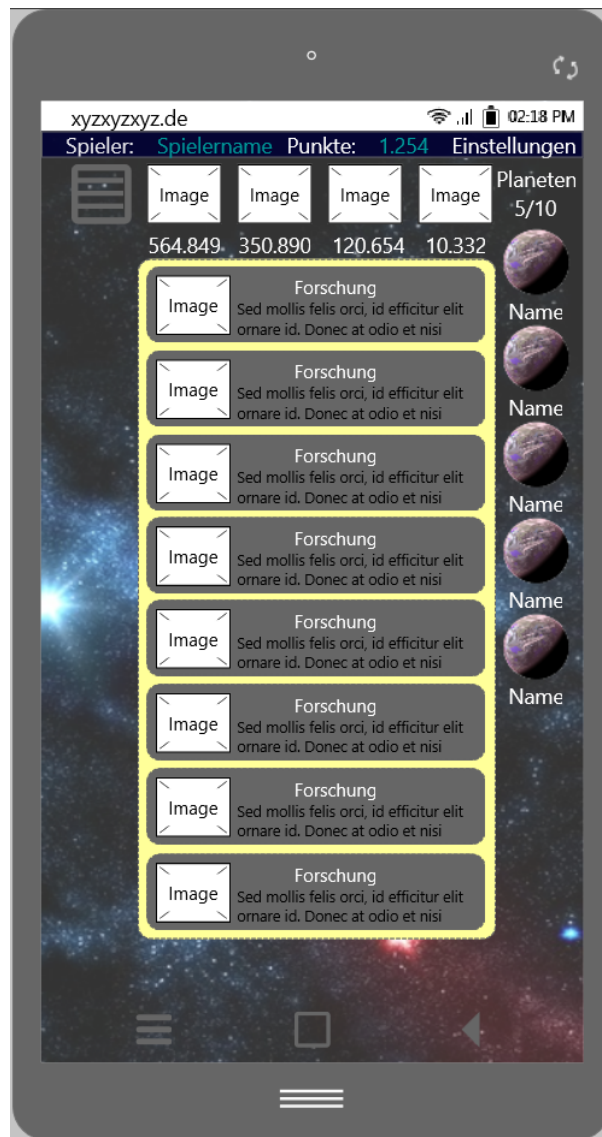


Bild 10: Mockup der Forschungsseite

Finales Konzept

Hauptsächlich wurden in den tatsächlichen Entwurf der Benutzeroberfläche Ideen aus den Mockups im [Portrait Layout](#) übernommen, da sich diese Darstellungsweise besser für Auflistungen eignet als das [Landscape Layout](#).

Der Startbildschirm wie er in *Bild 1* zu sehen ist wurde verworfen, da es innerhalb der Applikation kein Einstellungsmenü gibt. Damit verliert der Startbildschirm seine Funktion. Nun ist der Startbildschirm eine Übersichtsseite die der in *Bild 5* ähnlich sieht.

Auch das Dropdown Menü (*Bild 6*) wurde fast genau so übernommen. Die Planetenleiste am rechten Rand der Bilder mit [Portrait Layout](#) ist ganz weggefallen, da jeder Spieler nur einen Planet hat und nicht mehrere wie zu Anfang festgelegt. Die Seiten zur Anzeige der Gebäude (*Bild 9*) und Forschung (*Bild 10*) sind fast eins zu eins übernommen worden. Der einzige unterschied hier ist, das sie jetzt die gesamte Breite einnehmen.

Die Wahl der Farbe

Architekturbeschreibung

Implementierung

Spielkonzepte

Jeder Spieler kann auf seinem Planeten drei verschiedene Ressourcen abbauen, die für verschiedenste Gebäude und Verbesserungen benötigt werden. Der Abbau findet passiv statt. Das heißt, das auch wenn man nicht aktiv im Spiel - also "Offline" - ist werden Ressourcen erzeugt. Diese Ressourcen sind Baumaterialien, Computerchips und Treibstoff. Sie können verwendet werden, um Gebäude zu errichten, Raumschiffe und Verteidigungsanlagen zu bauen und Forschung zu betreiben. Des weiteren können Gebäude Verbessert werden, was verschiedene Boni mit sich bringt. Zum einen gibt es Gebäude, die die Produktion bestimmter Ressourcen verbessern, zum anderen gibt es Gebäude die die maximale Kapazität der Ressourcen erhöht. Des weiteren gibt es Gebäude die die Produktivität verschiedener Aspekte, wie zum Beispiel die Geschwindigkeit der Forschung oder das Bauen von Raumschiffen verbessert.

Die meisten Gebäude können dazu noch aufgewertet werden, was dazu führt das die Ressourcenkosten mit dem jeweiligen Bonus steigen. Das dient dem Zweck dem Spieler einen Sinn für das Erlangen von Ressourcen zu geben. Durch die Verbesserung von Verteidigungsanlagen kann der Spieler seine Chancen vergrößern Angriffe von anderen Spielern abzuwehren.

Angriffe sind ein weiterer Weg um Ressourcen zu erhalten. Der Spieler kann seine Raumschiffe zu einem anderen Planeten schicken und versuchen einem anderen Spieler Ressourcen zu stehlen. An dieser Stelle kommen die Verteidigungsanlagen, die man auf einem Planeten errichten und verbessern kann nun ins Spiel. Durch diese erhöht man seine Chance den Kampf zu gewinnen, da diese zusätzlich zu den eigenen Raumschiffen am Kampf teilnehmen.

Raumschiffe haben vier verschiedene Werte. Trefferpunkte, Schilde, Angriff und Feuerrate. Anhand dieser Werte wird ein Kampf simuliert. Die Verschiedenen Schiffe haben Stärken und Schwächen wie zum Beispiel eine besonders hoher Schaden aber dafür eine geringere Feuerrate oder ähnliches.

Tests und Usability

Usability

Benutzeroberfläche

Die Benutzeroberfläche wurde so entworfen, dass sich alle Funktionen entweder als Einträge im immer sichtbaren Dropdown Menü wiederfinden oder als Buttons an dem von der ausgeführten Aktion betroffenen Element sind. Des weiteren wurde für alle Seiten mit ähnlichem, Listenartigen Aufbau das gleiche Muster verwendet. Einzelne, zusammengehörende Elemente sind von einem Rahmen umgeben um sie voneinander abzugrenzen. Das Dropdown Menü beinhaltet fast ausschließlich Punkte zum Wechseln der Seite. Das sind Aktionen, die man nicht permanent benötigt und daher werden sie durch die Platzierung im Dropdown Menü versteckt. Dadurch kann Platz für die wichtigeren Aktionen geschaffen werden, die somit mehr ins Auge fallen.

Zusammenfassung

Soll-/Ist-Vergleich

Bei einer rückblickenden Betrachtung des Projektes, kann festgehalten werden, dass fast alle zuvor festgelegten [Anforderungen](#) erfüllt wurden. Außerdem wurden Aspekte erfüllt, die über die Planung hinaus gehen.

Lessons learned

Im Laufe des Projekts haben wir wertvolle Erfahrungen bezüglich der Entwicklung von Applikationen sammeln können. Des weiteren haben wir festgestellt das die Entwicklung eines Spieles eine größere Herausforderung ist, als wir Anfangs dachten. Außerdem haben wir viel neues über die Client Server Architektur, explizit die Kommunikation über HTTP sowie die Nützlichkeit von [Node.js](#) lernen können. [Node.js](#) regelt einen Großteil des handshakes von HTTP, sodass man sich nur noch um die Verarbeitung der Daten kümmern muss. Auch im Umgang mit einer Datenbank ist es von großem nutzen, da man Queries als Strings sehr einfach ausführen kann und man ein Objekt zurück bekommt das man direkt versenden kann. Auch haben wir von neuem gemerkt das das Dateiformat [JSON](#) die Kommunikation und Datenverarbeitung durch die Objektifizierung deutlich erleichtert.