

iTanky

iTanky ist ein kostenloses 2D Panzer Spiel für das iPhone. Die App ist zurzeit noch nicht im AppStore verfügbar, daher versuche ich in diesem Dokument die App so genau wie möglich zu beschreiben.



Idee

Ich war immer schon fasziniert von einfachen 2 dimensionalen Panzer Spielen, wo nicht die Grafik im Vordergrund stand, sondern der Spielspaß. Man kann einfach, ohne eine Anleitung lesen zu müssen, zu spielen beginnen. Eine wichtige Funktion für mich persönlich ist der Multiplayer Support. Spiele machen einfach viel mehr Spaß, wenn man gegen seine Freunde kämpfen kann. Das Ziel ist es, mit dem eigenen Panzer durch das Spielfeld zu fahren und die anderen Panzer zu besiegen. Dabei gibt es viele verschiedene Spielobjekte, wie Items, Minen, Wasser und sonstige Hindernisse.

Die ersten Konzepte

Die erste Version von iTanky sah in etwa so aus. Es wurden einfache Imageviews verwendet und es war nur eine sehr einfache Physik implementiert. Ich habe erst diesen Sommer Objective C gelernt (Die Programmiersprache von Apple für das iPhone und Mac OS), daher hat es einige Zeit gedauert, bis ich mich in die Spieleentwicklung unter iOS eingearbeitet habe. Seit September 2011 entwickle ich an iTanky.



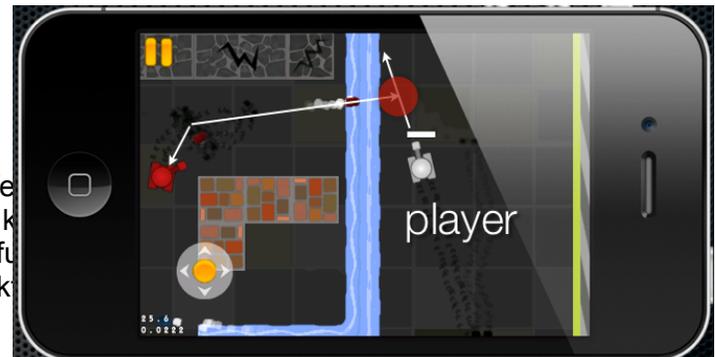
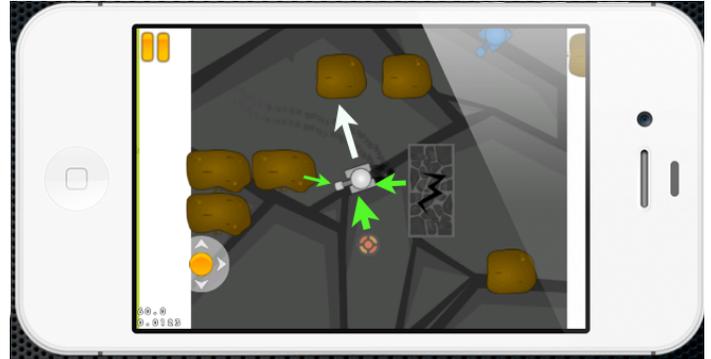
Umsetzung

Ich habe mich entschieden, das Cocos2d Framework zu benutzen, welches eine bessere Performance für 2D Spiele im Vergleich zu Apple's Standard UI-Elementen bietet. iTanky verwendet keine Physik Engine. Ich habe alle physikalischen Berechnungen (Prallen von Schüssen, Explosionen, Fahrverhalten, Kollisionen, etc.) komplett selbst programmiert. Als Programmiersprache habe ich nur Objective C verwendet.

Künstliche Intelligenz

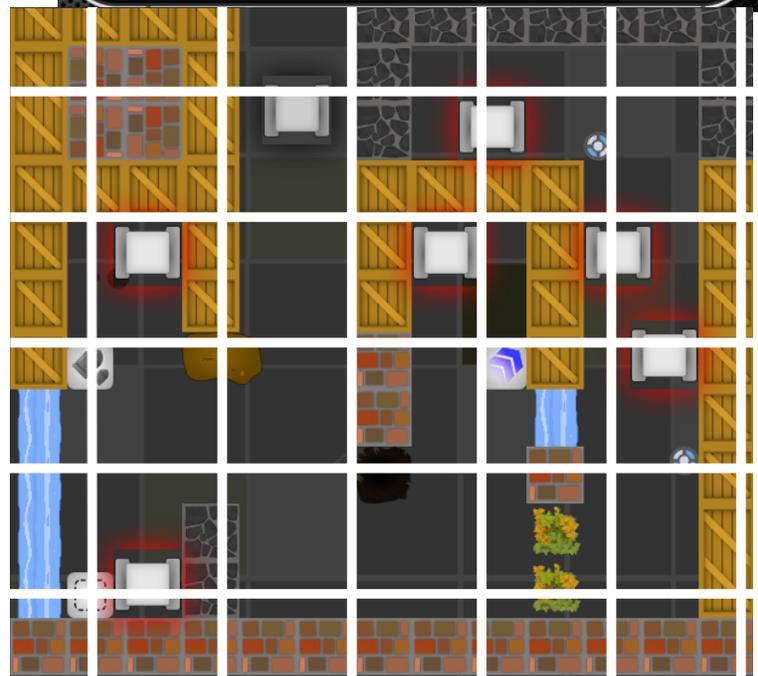
Da man im Kampagnenmodus gegen computergesteuerte Panzer spielt, habe ich auch eine künstliche Intelligenz programmiert. Diese erfüllt folgende Funktionen:

- **Automatisches Fahren:** Die Panzer fahren, ohne an Objekte anzustoßen. Dazu verwende ich eine Art verkehrte Gravitationkraft. Das bedeutet, dass z.B. Wände und Minen abstoßende Kräfte haben, während Items anziehend sind. Dadurch fährt der Panzer sehr sanft und flüssig. Grafik: Die grünen Pfeile sind abstoßend, der weiße ist die resultierende Beschleunigung
- **Verteidigen:** Die Panzer schauen, ob es gerade irgendwelche Schüsse gibt, die auf sie zufliegen und verteidigen sich automatisch.
- **Angreifen:** Die Panzer überprüfen, ob sie gerade auf den Spieler schießen können. Falls ja, rechnen sie sich genau aus, wo der Panzer des Spielers in x Sekunden sein wird. (siehe roter Punkt). Bei der höchsten Schwierigkeitsstufe wird man zu 100% getroffen. Geschwindigkeit ändert. Das bedeutet, die KI k... Spielfeld in x Sekunden aussieht. Das ganze fu... Berechnungen (keine Schleifen), dadurch funk...



Kollisionserkennung

Wie man auf diesem Bild sieht, können Karten ziemlich komplex und groß werden. Wenn man Kollisionen zwischen jedem Panzer, Block, Wasser, etc. kontrollieren würde, wäre die Performance sehr schlecht. Daher habe ich mich entschieden, eine Art Raster über die Map zu legen. Ich merke mir, in welchem Raster sich welche Objekte befinden und muss daher nur noch wenige Kontrollen machen. (Der Raster ist natürlich nur für die einfache Vorstellung, im Code sieht das Ganze komplett anders aus)



Multiplayer

Der Mehrspielermodus ist eine der wichtigsten Funktionen dieses Spieles. Ich verwende das Apple Framework "GameKit" für die Netzwerkkommunikation über Bluetooth. GameKit hilft bei der Verbindungsherstellung mit

einem anderen iPhone, danach muss die komplette Netzwerksynchronisation händisch programmiert werden.

Der typische Ablauf eines Multiplayerspieles:

- **Verbinden:** Überprüfung, ob App-Version kompatibel ist, Entscheidung wer der Host ist
- **Übertragen:** Übertragen der ausgewählten Panzer, Entscheidung der Farben, Senden der Karte
- **Abgleichen:** Pakete müssen bestätigt werden (TCP), Warten auf Kalibrierung (falls Acceleration Steuerung aktiviert ist), Warten bis die Karte auf beiden Geräte erfolgreich geladen worden ist
- **Starten:** Das Spiel wird zeitgleich auf beiden Geräten gestartet und Countdown wird angezeigt
- **Spielen:** Das eigentliche Spiel. Übertragen werden über **20** verschiedene Arten von Nachrichten, wie z.B. tank_move, tank_hit, mine_explode, etc. mit jeweils verschiedenen Eigenschaften und Funktionen. Das dient zur Trafficoptimierung, da immer nur die benötigten Informationen gesendet werden. Hier wird je nach Art der Nachricht TCP oder UDP verwendet.
- **Ende:** Entscheidung wer gewonnen hat, Beenden des Spiels

Die Fehlerbehandlung und Korrekturen von Race Conditions waren auch ein großer Teil des Programmieraufwands vom Multiplayermodus.

Map Editor

Da die iPhone Apps mit vielen vorinstallierten Level ausgeliefert werden soll, habe ich eine Möglichkeit gebraucht, einfach und schnell neue Karten zu entwerfen und speichern.

Daher habe ich die Webseite <http://maps.itanky.com> programmiert.

Die Logindaten zum Testen sind:

Nutzername: "u19" (klein geschrieben)

Passwort: "temp"

Zuerst wählt man auf der linken Seite eine Kategorie aus, wie z.B. Panzer oder Wände.

Danach wählt man ein Objekt per Klick aus.

Sobald man auf die Karte klickt erscheint das ausgewählte Objekt auf dieser Stelle.

Mittels Drag&Drop können die Elemente beliebig verschoben werden. Die Karte kann rechts unten auch per Ziehen vergrößert und verkleinert werden. Sobald die Karte fertig ist, kann man die Karte auch speichern.

Dazu sucht man sich einen passenden Namen aus und wählt das gewünschte Design aus. Im Nachhinein kann jederzeit das Design oder die Karte selbst verändert und erneut gespeichert werden. Wenn man auf "Approve" klickt, wird die Level als "Fertige Karte" gespeichert und automatisch beim App-Start von iTanky heruntergeladen.



Es gibt 3 verschiedene Designs, wo dann auch die Blöcke im Spiel dem Design entsprechend aussehen.

Man kann 3 verschiedene **Panzer** verwenden, wobei der Spieler selbst seinen aussuchen kann. Der Panzer vom Spieler wird mit einem blauen Schatten hinterlegt.

Es gibt viele verschiedene **Wände** wie z.B. Holz, Stein, Ziegelsteine, etc

Es gibt 4 verschiedene **Untergründe**, die in beliebiger Größe auf das Feld positioniert werden können. Alle haben natürlich unterschiedlichen Eigenschaften die sich auf das Spiel auswirken. Zurzeit stehen dem Spiel 7 verschiedene **Items** zur Verfügung, es sind aber noch einige für die Zukunft geplant.

Zusätzlich können noch **Minen** auf das Feld gelegt werden, die explodieren, sobald ein Gegner in die Nähe kommt. Die rote Mine explodiert und beschädigt die Panzer in der Nähe. Die blaue Mine beschädigt auch, bremst (friert) die Panzer aber zusätzlich für eine gewisse Zeit.

Der Editor funktioniert auf allen gängigen Browser, ist aber für Safari/Chrome für Mac optimiert.

Shortcuts:

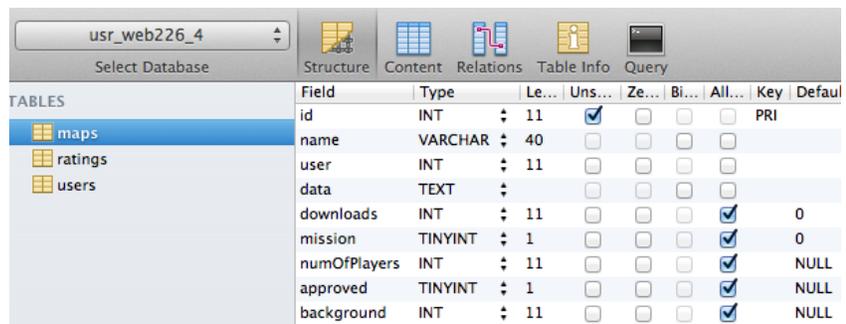
- d: Delete object
- r: rotate object
- 1: set tank as player tank
- Pfeiltasten um Objekte genau zu bewegen

Technische Umsetzung des Editors

Den Editor habe ich mit HTML, CSS, Javascript, jQuery, AJAX, CSS3, PHP, MySQL, JSON und jQuery UI programmiert. Das Laden der Maps funktioniert mittels AJAX, daher muss die Seite nicht nach jeder Kleinigkeit aktualisiert werden. Gespeichert werden die Elemente vom DOM im JSON-

Format in einer MySQL-Datenbank. Das Menü am Rand wird mit jQuery animiert. Sobald eine Kategorie aufgeklappt wird, klappt sich eine andere zu. Die komplette Webseite ist dynamisch generiert und daher einfach zu erweitern. CSS3 wird unter Anderem für die Schlagschatten von Objekten und den Informationsfenster (wenn man über ein Element im Menü drüberfährt) verwendet.

Die iPhone app verwendet zum Herunterladen der Karten eine Schnittstelle, welche ich in PHP programmiert habe: <http://maps.itanky.com/download.php> (das sind alle zugelassenen Karten)



Field	Type	Le...	Uns...	Ze...	Bl...	All...	Key	Defau
id	INT	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI	
name	VARCHAR	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
user	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
data	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
downloads	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
mission	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
numOfPlayers	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL
approved	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL
background	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL

Menü

Das **Hauptmenü** hat folgende Buttons

- Kampagne
- Mehrspielermodus
- Info-Button für Links auf iTanky-Webseite und Facebook Seite

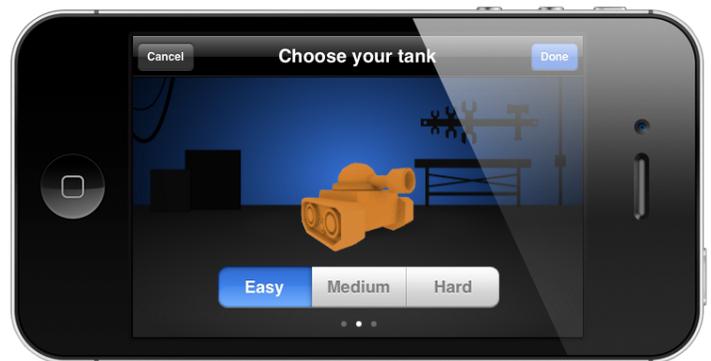


- Einstellungs-Button für Soundeinstellungen und Controlleroptionen

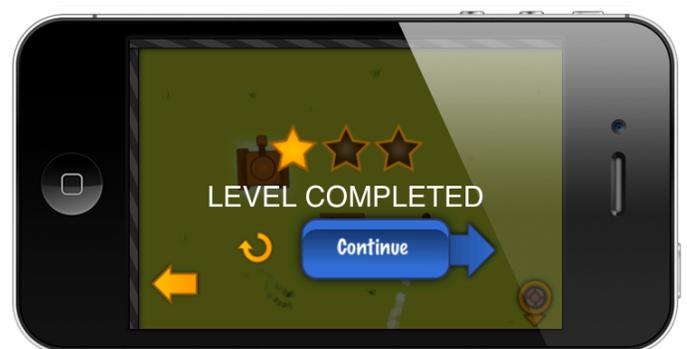
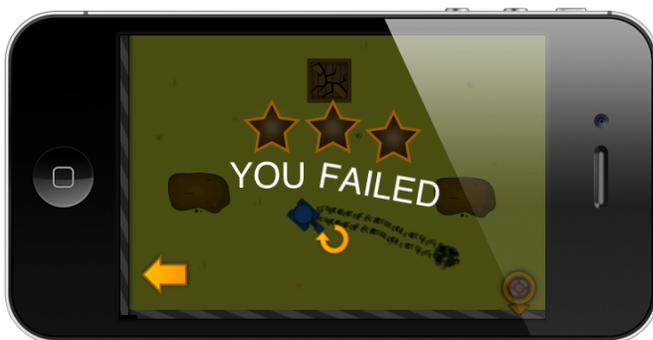
Die **Levelauswahl** listet alle geladenen Karten in einem übersichtlichen Menü auf. Man sieht sofort, welches Thema die jeweilige Map hat. Die Sterne symbolisieren, in welchem Schwierigkeitsgrad man die Level bereits abgeschlossen hat. In der finalen Version ist pro Seite nur ein Design. Unten sieht man an den Punkten auf welcher Seite man sich gerade befindet. Man kann neue Level freischalten. Die Scores und Karten werden in einer CoreData Datenbank gespeichert.



In der **Panzerwahl** kann man den eigenen Panzer durch "Wischen" am Bildschirm auswählen. Man sieht ein 3D-Modell des jeweiligen Panzers. Direkt darunter kann der Schwierigkeitsgrad im Einzelspielermodus ausgewählt werden.



Im **Endbildschirm** wird angezeigt, ob man gewonnen hat und wie viele Sterne man bekommen hat.



Technische Details

Die App verwendet CoreData für die Speicherung der Karten. Daher können diese sehr dynamisch ergänzt und erweitert werden. Beim ersten App Start werden automatisch alle zugelassenen ("approved") Karten in dem JSON-Format heruntergeladen.

Für das Anzeigen und Rendern der Objekte am iPhone verwende ich CCSprites vom Cocos2d Framework. Diese habe ich um über 40 neue Methoden erweitert, die den Zugriff deutlich erleichtern und beschleunigen. Außerdem habe ich im CCFollow die Kamera Funktion umgeschrieben, damit Kamerabewegungen sanft und flüssig erfolgen.

Die iPhone App besteht aus über **5000 Zeilen Code** und der Level Editor aus über 2000. Für die Explosionseffekte habe ich die eingebauten Partikeleffekte verwendet, für die Spuren, Rauch, etc. habe ich meine eigene, schnellere Engine programmiert.

Folgende Programme wurden zum Erstellen dieser App verwendet:

- Xcode: für die Programmierung der App
- Espresso: für die Programmierung des Level Editors
- iOS Simulator: zum Testen und Debuggen der App
- Textmate: für SQL-Dumps, Plist files und diverse andere Files
- Sequel Pro: für die MySQL Datenbank der Maps
- GitX in Kombination mit bitbucket.org: für die Sourcecodeverwaltung
- Dropbox: für das Sichern und Freigeben des Projekts
- Testflight: für das Senden von Beta Versionen an Freunde und Kollegen
- Wunderkit: für Tasks und Notizen
- Pixelmator: für das Erstellen und Bearbeiten von Grafiken
- Sound Stage: für das Aufnehmen der Bildschirmvideos für den Trailer

Grafiken

Die Grafiken wurden von Richard Forstmann erstellt. Er hat schon einige Animationsfilme selbst entwickelt und war einer der Gewinner beim MB21 Wettbewerb, wo ich ihn auch kennen gelernt habe. Richard ist erst 14 Jahre alt und ist ein sehr begabter Designer und Grafiker. Er kommt aus Dresden, Deutschland und wurde deswegen nicht als Künstler eingetragen. Wir haben über Skype, Email und Wunderkit kommuniziert und zusammengearbeitet. Richard hat Zugriff auf die Grafiken per Dropbox, stimmen die Dateinamen immer überein.

Zukunft

- **Öffentlicher Level Editor** (zurzeit passwortgeschützt)
- **Spezialmissionen** (z.B: sammle alle Kisten, ohne beschädigt zu werden)
- **Pro Version** (kostenpflichtige Version mit mehr Funktionen)
- **Game Center Achievements**
- **Level community** (Man soll sich registrieren können, um selbst Karten bauen zu können. Am iPhone kann man sich die Level anderer Nutzer anschauen, herunterladen, bewerten und kommentieren. Es soll eine Art Level-Community entstehen)

Links

iTanky Webseite: <http://itanky.com>
iTanky Map Editor: <http://maps.itanky.com>
Facebook Seite: <http://facebook.com/iTankyApp>
Twitter: <http://twitter.com/krausefx>