Konzeptbericht

Status	**In Arbeit** / In Prüfung / Abgeschlossen
Projektname	<CGR>
Projektleiter	<Ilija Tovilo>
Auftraggeber	<Martin Frieden(Gibb)>
Autoren	<Lukas Seglias> <Ilija Tovilo>
Verteiler	<-->

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
0.1	26.02.2013	Beschreiben der Use-Cases	Projektleiter/Lukas Seglias
0.1.1	05.03.2013	Beschreibung der Benutzerschnittstelle	Projektleiter/Lukas Seglias
0.1.2	09.03.2013	Beschreibung der Benutzerschnittstelle	Lukas Seglias
0.1.3	09.03.2012	Beschreibung der Systemarchitektur	Projektleiter
1.0	19.03.2013	Beschreibung der Klassen	Projektleiter/Lukas Seglias

Referenzen

Referenz	Titel, Quelle
[1]	Buch „Hermes Grundwissen“

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Zusammenfassung

In diesem Dokument wird genau beschrieben, welche Anwendungsfälle für das gegebene Projekt existieren. Es wird genau beschrieben, wie die Applikation umgesetzt wird, was diese unterstützten soll.

Es wird erstmalig gezeigt, wie das Endprodukt aussehen wird.
Das Datenmodell, die Schnittstellen und das Klassendiagramm werden definiert.

2 Systemanforderungen

2.1 Übersicht Anwendungsfälle

Der User interagiert ausschliesslich mit dem iPhone. Diese Interaktionen werden an das Spiel weitergeleitet und verarbeitet.
Dies ist das dazugehörige Use-Case Diagramm.

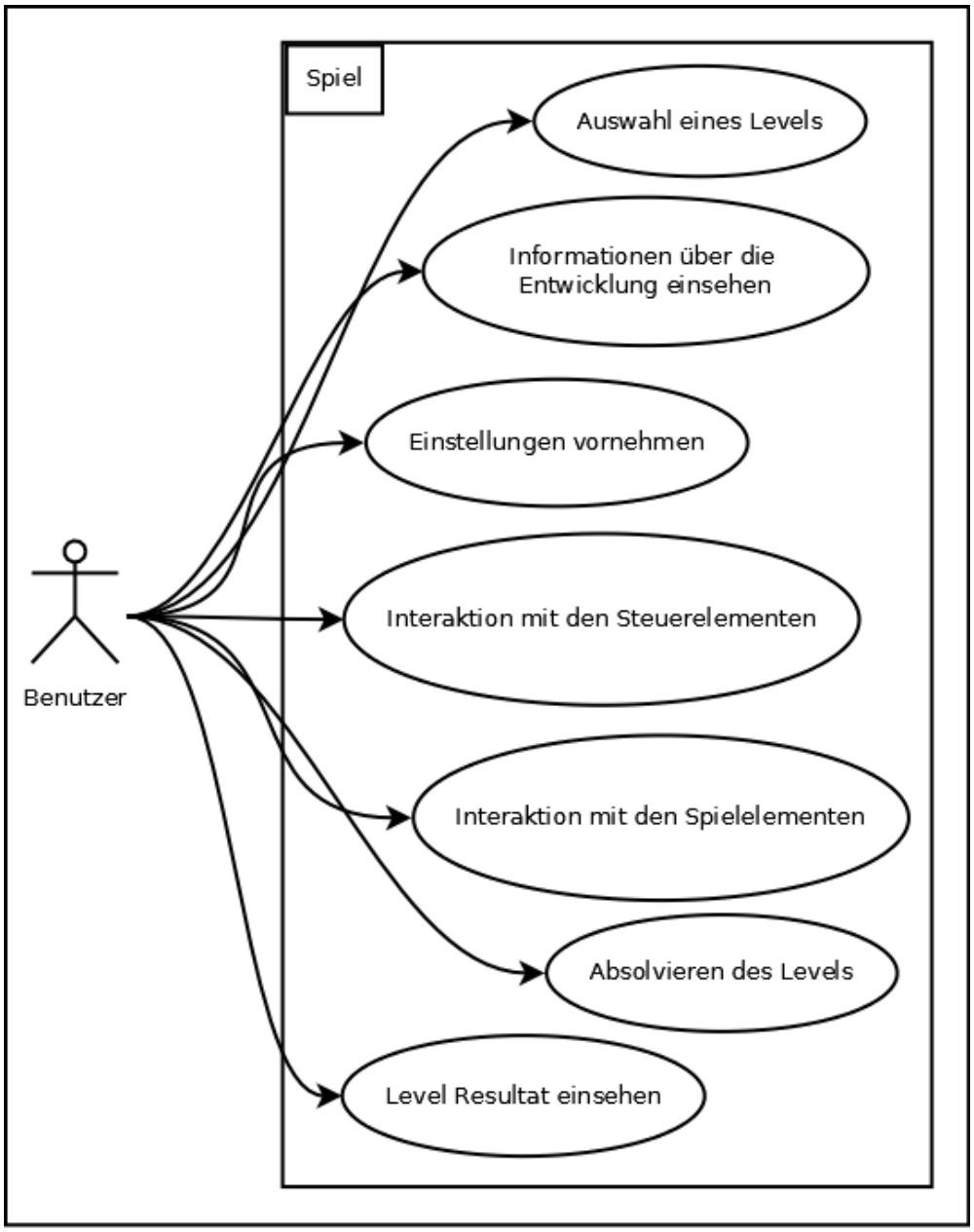

Abbildung 1: Use-Cases

2.2 Anwendungsfälle

2.2.1 Auswahl eines Levels

Anwendungsfall „Auswahl eines Levels“	
Kurzbeschreibung	Der Benutzer kann im Spiel selber auswählen, welchen Level er gerne spielen möchte. Er kann diese auswählen wenn er im Hauptmenu auf „Play“ drückt. Er erhält eine Liste aller Welten, und in den Welten eine Liste der dazugehörigen Levels.
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat das Spiel auf seinem iPhone installiert und gestartet
Ablauf	1. Der Benutzer startet das Spiel und drückt auf „Play“ 2. Der Benutzer wählt eine der freigeschalteten Welten 3. Der wählt einen der freigeschalteten Level
Resultat	Der Level wird gestartet, worauf der Benutzer mit dem Spielen beginnen kann.
Ausnahmen	-

2.2.2 Informationen über die Entwicklung einsehen

Anwendungsfall „Informationen über die Entwicklung einsehen“	
Kurzbeschreibung	Es gibt im Hauptmenu einen Punkt „Info“, der als Fragezeichen dargestellt ist. Hier sieht der Benutzer den Link auf unsere Facebook/Twitter Seite und kann uns liken/folgen. Er sieht ebenfalls wer für das Projekt verantwortlich ist.
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat das Spiel auf seinem iPhone installiert und gestartet
Ablauf	1. Der Benutzer wählt im Hauptmenu den Punk „Info“ 2. Der Benutzer drückt auf den Facebook/Twitter Knopf
Resultat	Der Benutzer kann uns unterstützten, falls ihm unser Projekt gefällt.
Ausnahmen	-

2.2.3 Einstellungen vornehmen

Anwendungsfall „Einstellungen vornehmen“	
Kurzbeschreibung	Der Benutzer kann im Spiel Einstellungen vornehmen. Er kann hier lediglich den Ton ein/aus-schalten und Musik ein/aus-schalten.
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat das Spiel auf seinem iPhone installiert und gestartet
Ablauf	1. Der Benutzer wählt den Punkt „Einstellungen“ 2. Er nimmt die gewünschten Änderungen vor
Resultat	Der Benutzer kann das Spiel wie gewünscht konfigurieren
Ausnahmen	-

2.2.4 Interaktion mit den Steuerelementen

Anwendungsfall „Interaktion mit den Steuerelementen“	
Kurzbeschreibung	Der Benutzer kann Mario mit den Steuerelementen steuern
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat ein Level ausgewählt
Ablauf	1. Der Benutzer kann mit dem Steuerknüppel Mario nach links/rechts steuern, oder nach unten ducken lassen 2. Der Benutzer kann Mario mit dem B-Knopf a. schnell rennen lassen b. Feuerbälle spucken lassen 3. Der Benutzer kann Mario mit dem A-Knopf springen lassen
Resultat	Der Benutzer steuert Mario durch die herausfordernden Levels
Ausnahmen	-

2.2.5 Interaktion mit den Spielelementen

Anwendungsfall „Interaktion mit den Spielelementen“	
Kurzbeschreibung	Der Benutzer kann Mario steuern mit den Steuerelementen
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat ein Level ausgewählt
Ablauf	1. Mario weicht Gegnern aus, indem er über sie springt 2. Mario duckt sich vor feindlichen Geschossen 3. Mario springt auf Koopas, um diese zu töten und deren Panzer fallen zu lassen a. Er springt auf deren Panzer, um ihn als Geschoss einzusetzen 4. Mario springt auf Gumbas, wodurch diese sterben 5. Lakitus fliegen auf einer Wolke und werfen Spiny-Gegner herunter 6. Mario kann nicht auf den Kopf eines Spinys springen, da dieser einen Stachel besitzt 7. Mario kann Blöcke zerstören, indem einen Sprung unter diesen ausführt und sie mit der Faust zerschlägt 8. Mario kann Items aufnehmen a. Der Pilz verwandelt den kleinen zum grossen Mario b. Die Feuerblume verwandelt den kleinen/grossen Mario zum Feuerblumen-Mario c. Münzen erhöhen den Punktestand des Spielers
Resultat	Der Benutzer navigiert durch das Level
Ausnahmen	-

2.2.6 Absolvieren des Levels

Anwendungsfall „Absolvieren des Levels“	
Kurzbeschreibung	Der Benutzer absolviert das Level
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat ein Level ausgewählt
Ablauf	1. Der Benutzer gelangt an die Fahnen-Stange am Ende des Levels
Resultat	Der Benutzer absolviert das Level
Ausnahmen	-

2.2.7 Level-Resultat einsehen

Anwendungsfall „Level-Resultat einsehen“	
Kurzbeschreibung	Der Benutzer sieht nach dem Spielen des Levels seine Ergebnisse
Akteure	Benutzer
Vorbedingungen	Der Benutzer hat ein Level durchgeführt (gewonnen/verloren)
Ablauf	1. Der Benutzer hat ein Level durchgeführt 2. Er sieht seine a. Punkte-Zahl b. Benötigte Zeit c. Wiederholen-Knopf d. Level-Auswahl-Knopf 3. Falls gewonnen, sieht er den „Nächster Level“-Knopf
Resultat	Der Benutzer absolviert das Level
Ausnahmen	-

3 Benutzerschnittstelle

3.1 Start

Diese Szene wird beim Start des Spieles angezeigt.
Mit dem Play-Knopf kann zur Welten-Auswahl gewechselt werden.
Mit dem Zahnrad-Knopf kann zu den Einstellungen gewechselt werden.
Mit dem Fragezeichen-Knopf kann zu den Infos gewechselt werden.

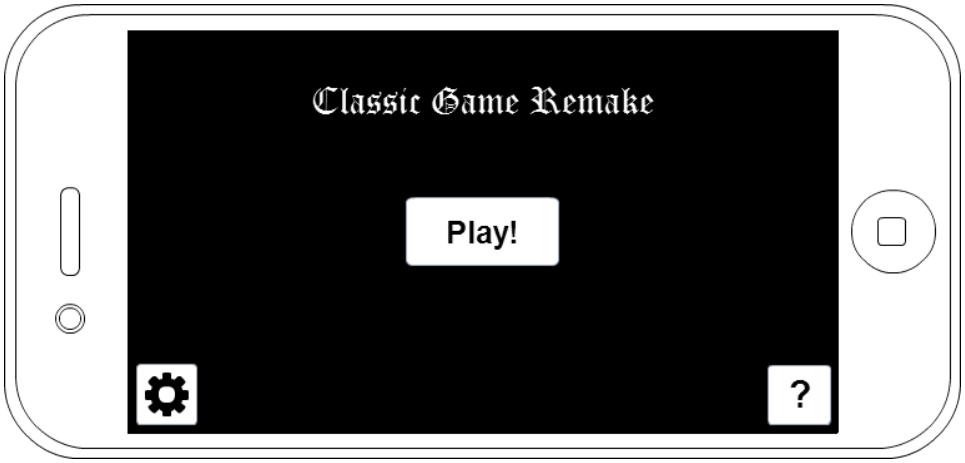

Abbildung 2: Die Startszene des Spiels

3.2 Infos

Hier erhält der Benutzer grundlegende Informationen über das Spiel, und über dessen Entwickler.
Er erhält auch Links zu den Social-Media Seiten des Spiels. Über den Menu-Knopf gelangt er wieder zur Start-Szene.

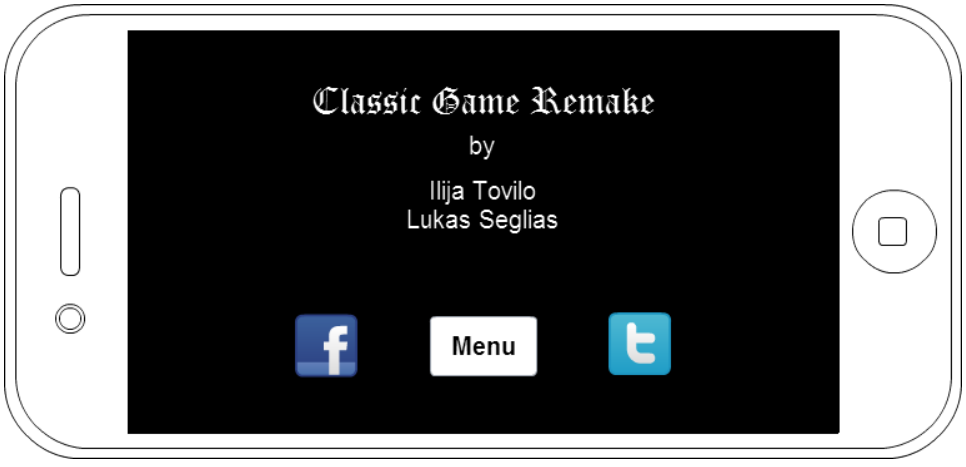

Abbildung 3: Informationen über die Entwickler des Spiels sind hier ersichtlich

3.3 Einstellungen

In den Einstellungen kann der Benutzer die Musik und die Töne des Spiels ein- oder ausschalten. Mit den Tönen sind die Geräusche, welche Spielelemente (Gegner, der Spieler selbst) erzeugen, gemeint. Die Änderung einer Einstellung tritt sofort in Kraft, es ist kein Neustart des Spiels notwendig. Durch Drücken des Menu-Knopfes kann der Benutzer wieder zur Start-Szene wechseln.

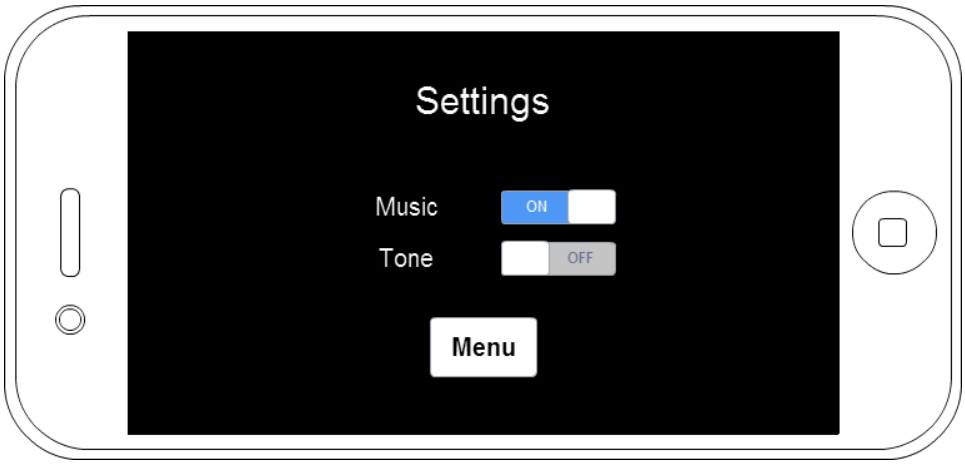

Abbildung 4: Die aktuellen Einstellungen können hier verändert/eingesehen werden.

3.4 Welten-Auswahl

In der Welten-Ansicht kann der Benutzer zwischen verschiedenen Welten auswählen, welche verschiedene Levels enthalten. Eine solche Welt kann durch erfolgreiches Absolvieren anderer Welten freigeschaltet werden.

Die Punkte im unteren Teil der Szene sind Markierungen welche dem Benutzer zeigen, auf welcher Seite er sich gerade befindet und wie viele Seiten vorhanden sind. Durch ein Wischen nach Links gelangt er auf die zweite Seite und erhält eine Übersicht anderer Welten. Durch Drücken des Pfeil-Knopfes gelangt er wieder zur Start-Szene.

Sobald der Benutzer eine Welt ausgewählt hat, gelangt er zu deren Level-Auswahl.

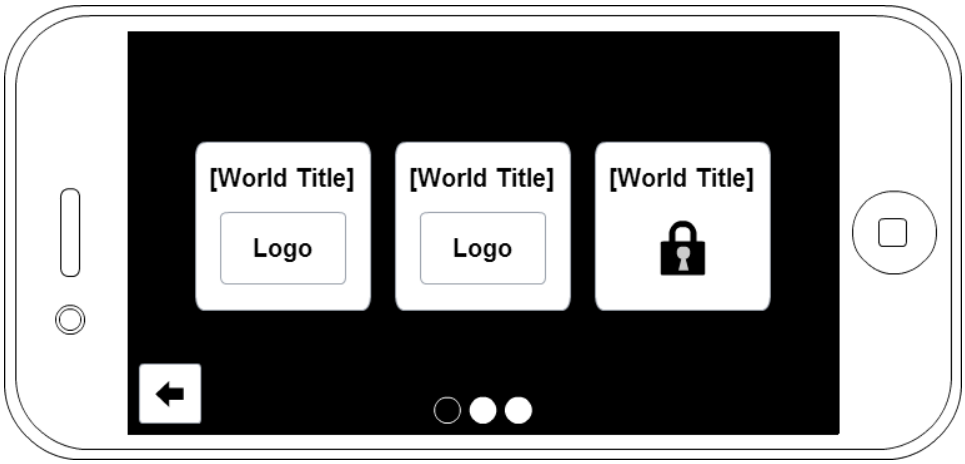

Abbildung 5: Die Ansicht aller Welten.

3.5 Level-Auswahl

In dieser Ansicht sind alle Levels der ausgewählten Welt einsehbar. Die Levels sind aufsteigend nummeriert. Ein Schloss-Symbol auf einem Level bedeutet dass der vorangehende Level absolviert werden muss, um diesen Level freizuschalten.

Wie auch in der Welten-Auswahl ist die Level-Auswahl auf mehrere Seiten aufgeteilt.

Über den Pfeil-Knopf gelangt der Benutzer zurück zur Welten-Auswahl. Sobald er einen freigeschalteten Level auswählt, wird das Spiel in diesem Level gestartet und die Ingame-Ansicht wird angezeigt.

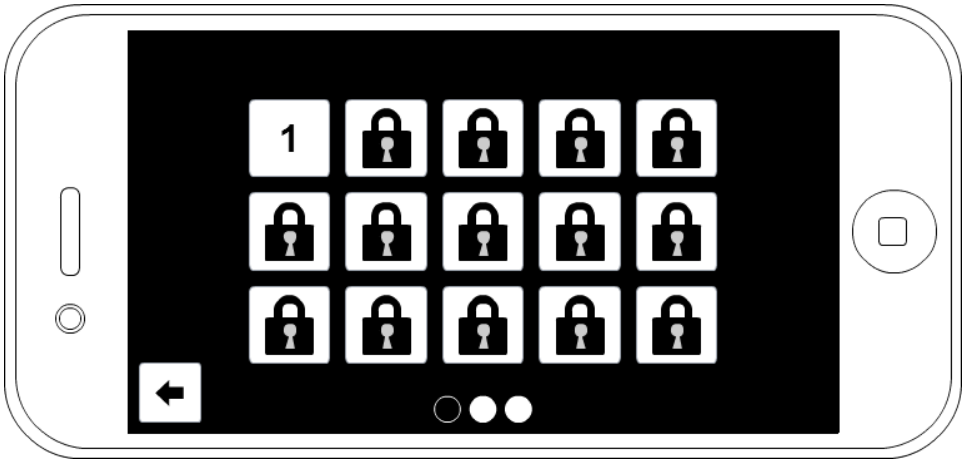

Abbildung 6: Die Auswahl eines freigeschalteten Levels.

3.6 Im Spiel

Sobald der Level ausgewählt wurde, startet das Spiel. In der oberen linken Ecke ist der Pause-Knopf. Durch dessen Betätigung wird zur Pause-Szene gewechselt. Im unteren rechten Bereich sind der A- und B-Knopf anzutreffen. Durch Betätigen des A-Knopfes springt Mario. Durch Betätigen des B-Knopfes beschleunigt Mario oder wirft Feuerbälle, sofern er eine Feuerblume gesammelt hat.

Unten links ist das Control Pad. Dieses wird für die Bewegung in eine beliebige Richtung verwendet. Wenn das Control Pad nach unten gezogen wird, duckt sich Mario. Das Ziehen nach links und rechts bewirk, dass sich Mario in die entsprechende Richtung bewegt. Das Ziehen nach oben hat keine Wirkung.

Sobald Mario ans Ende eines Levels gelangt und an der Fahnenstange herunterrutscht, wird die „Erfolgreicher Level-Abschluss“-Szene angezeigt. Falls Mario auf der Strecke von Gegnern getötet wird, oder in eine Grube fällt, wird die „Fehlgeschlagener Level-Abschluss“-Szene angezeigt. Falls der Level nicht in der vorgegebenen Zeit absolviert wird, ist das Spiel vorbei.
Beim Sammeln von Münzen und bezwingen von Gegnern erhält der Spieler Punkte.

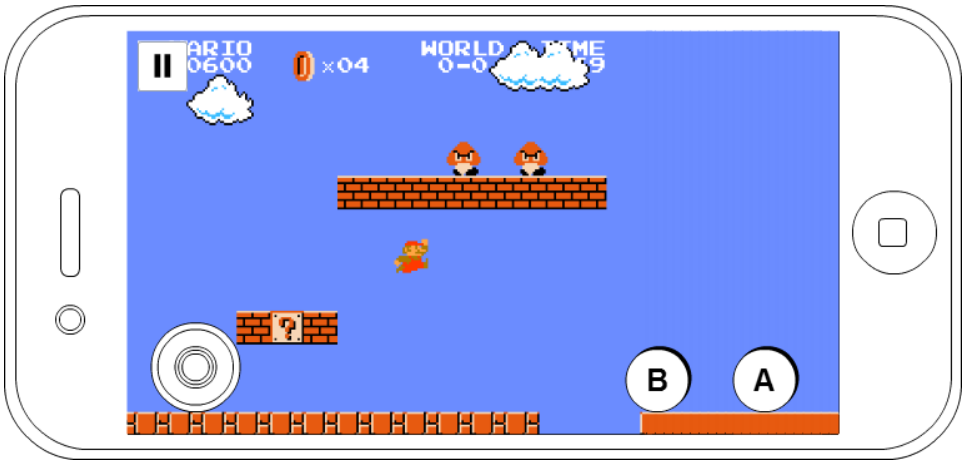

Abbildung 7: Ingame-Ansicht

3.7 Pause

In dieser Szene ist das Spiel pausiert. Im oberen linken Bereich ist die Bezeichnung des aktuellen Levels zu sehen. Diese Bezeichnung ist in folgendem Format definiert: Weltnummer-Levelnummer. Falls der Spieler auf den Play-Knopf drückt, wird das Spiel fortgesetzt (Die Ingame-Ansicht wird angezeigt).

Mithilfe des Levels-Knopfes beendet der Benutzer den aktuellen Level und gelangt zur Level-Auswahl zurück. Der Retry-Knopf startet den aktuellen Level nochmals von vorne.

Abbildung 8: Die Ansicht des pausierten Spieles.

3.8 Erfolgreicher Level-Abschluss

Hier sieht der Benutzer seinen Punktestand und die Zeit in der er den Level absolviert hat. Mithilfe des Levels-Knopfes gelangt er zur Level-Auswahl zurück. Der Retry-Knopf startet den aktuellen Level nochmals von vorne. Beim Druck auf den Next-Knopf wird das Spiel direkt im nächsten Level gestartet.

Abbildung 9: Der Spieler wird beglückwünscht.

3.9 Fehlgeschlagener Level-Abschluss

Hier sieht der Benutzer seinen Punktestand und die Zeit nach der er gestorben ist. Mithilfe des Levels-Knopf gelangt er zur Level-Auswahl zurück. Der Retry-Knopf startet den aktuellen Level nochmals von vorne.

Abbildung 10: Leider hat der Spieler nicht gewonnen.

4 Systemarchitektur

4.1 Gliederung der Lösung

Unser Spiel hat konkret das iPhone als Ziel-Plattform.
Die Entwicklung von Spielen ist, abgesehen von Web-Sprachen, ausschliesslich mit OpenGL möglich.
Eine sehr gute Game-Engine, welche die Arbeit mit OpenGL abnehmen kann, ist Cocos2d.
Mehr ist dazu im Voranalyse-Bericht zu finden.

4.1.1 Schichten/Layers

Hier ein grober Überblick von der Cocos2d Game-Engine:

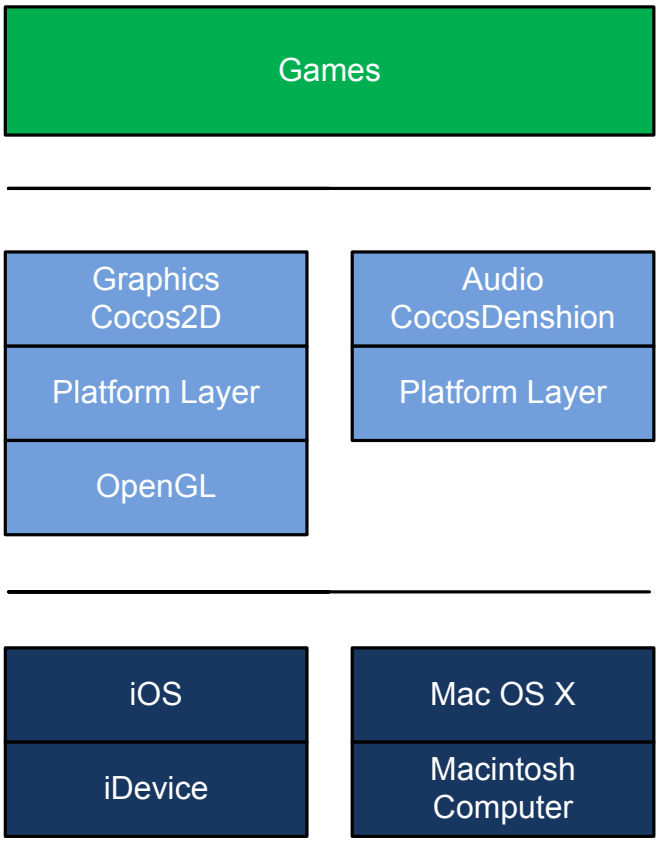

Abbildung 11: Struktur von Cocos2d

Wie hier ersichtlich ist, ist die Programmierung für iOS und Mac OS X bezüglich der Cocos2d Game-Engine identisch. Wir werden uns jedoch ausschliesslich auf die iOS Programmierung konzentrieren.

4.1.2 Pakete

Die Gliederung des Spiels ist sehr einfach. Es gibt keine Anbindung an einen Server, noch gibt es Benutzer-Daten, welche angelegt werden.
Wir benutzen die Open-Source Game-Engine Cocos2d.
Folglich gibt es zwei Pakete in unserem Projekt:

- Cocos2d
- Game

4.1.3 Klassen

Nachfolgend sind die Klassendiagramme unseres Spiels. Ausserdem sind die Klassen, welche als Schnittstelle zu Cocos2d genutzt werden, aufgeführt. Alle Klassen von Cocos2d haben den Präfix „CC“. Unsere eigenen Klassen haben den Präfix „ST“.

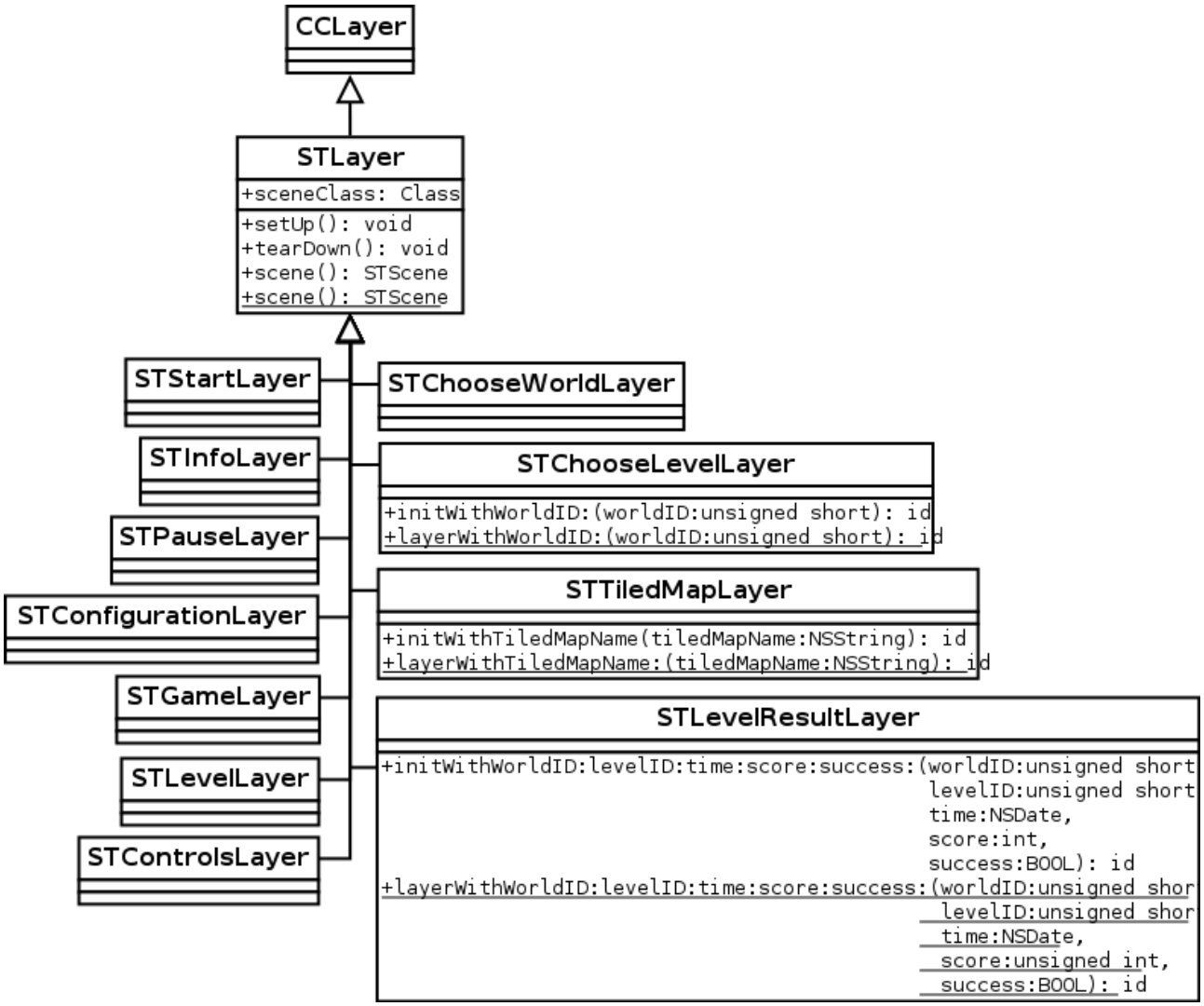

Abbildung 12: Layer-Klassen

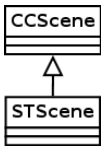

Abbildung 13: Scene-Klassen

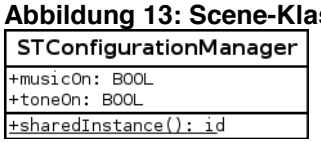

Abbildung 14: Einstellungen

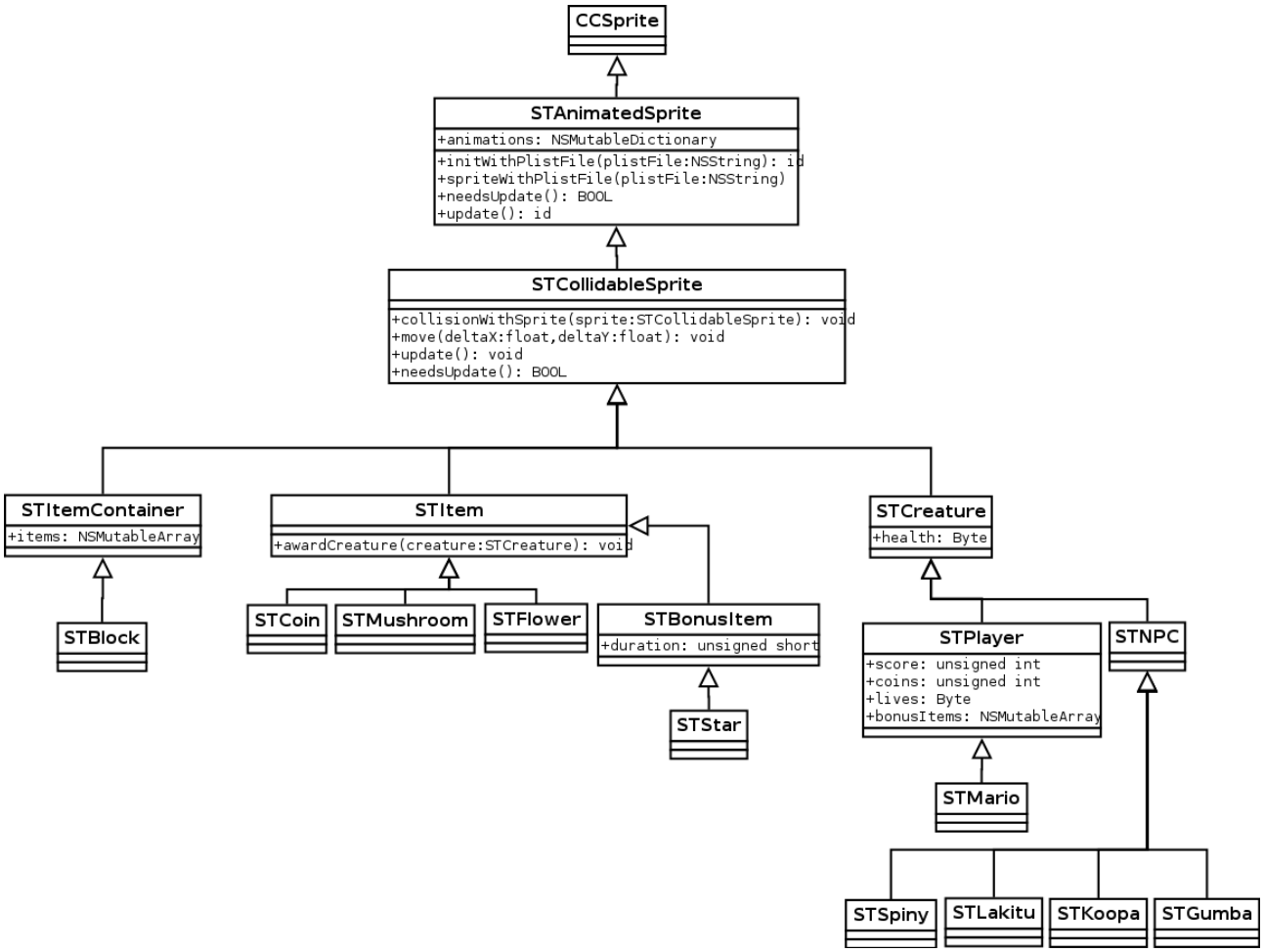

Abbildung 15: Sprite-Klassen

4.1.4 Module

Unsere Programmierung erfolgt objekt-orientiert, weshalb durch das Klassen-Diagramm alle Fragen geklärt sein sollten.

4.2 Beschreibung der Elemente

4.2.1 STLayer

STLayer ist eine Subklasse von CCLayer.

Die CCLayer-Klasse ist eine vorgegebene Klasse von Cocos2d.
Ein Layer ist, wie der Name schon sagt, eine Schicht. Um genau zu sagen eine Schicht einer Szene, welche schlussendlich auf dem Bildschirm präsentiert wird.

Hier können spezifischere Funktionen implementiert werden.
Zum Beispiel kann ein STLayer-Objekt sich gleich in ein STScene-Objekt einbetten lassen.
Dies ist momentan der einzige Vorteil.

4.2.1.1 STStartLayer

Auf diesem Layer wird das Start-Menu präsentiert.
Hier hat der Benutzer die Option, das Spiel zu starten, Infos anzusehen, Einstellungen vorzunehmen.

4.2.1.2 STInfoLayer

Auf diesem Layer werden die Infos zu den Entwicklern angezeigt.

4.2.1.3 STTiledMapLayer

Diese Klasse kann sogenannte Tiled Maps darstellen.

4.2.1.4 STGameLayer
Dieser Layer beinhaltet Sub-Layers für das schlussendliche Game-Play.
Sub-Layers:
- STLevelLayer
- STControlLayer
- STPauseLayer

4.2.1.5 STLevelLayer
Dieser Layer zeigt alle Spielelemente an. Zum Beispiel Mario, alle Blöcke, alle Gegner usw.
Deshalb ist dieser Layer eine Sub-Klasse der STTiledMapLayer-Klasse.

4.2.1.6 STPauseLayer
Dieser Layer wird angezeigt, wenn das Spiel

4.2.1.7 STControlLayer
In diesem Layer werden die Controls zum steuern des Spiels angezeigt.

4.2.1.8 STConfigurationLayer
In diesem Layer werden die Controls zum vornehmen der Einstellungen angezeigt.

4.2.1.9 STChooseWorldLayer
In diesem Layer werden die Welten, welche zur Auswahl stehen, aufgelistet.
Beim auswählen einer dieser Welten werden die spezifischen Levels dieser Welt mithilfe der STChooseLevelLayer-Klasse aufgelistet.

4.2.1.10 STChooseLevelLayer
In diesem Layer können die spezifischen Levels einer Welt ausgewählt werden.
Beim auswählen eines Levels wird dieses schliesslich mit Hilfe der STGameLayer-Klasse gestartet.

4.2.1.11 STLevelResultLayer
In diesem Layer werden die Resultate nach Abschluss eines Levels angezeigt.
Der Benutzer kann den Level wiederholen. Falls der Versuch geglückt ist, kann der Benutzer gleich zum nächsten Level wechseln.

4.2.2 STScene
Die STScene-Klasse ist eine Sub-Klasse von CCScene. Eine CCScene kann schlussendlich als einziges Objekt auf dem Bildschirm präsentiert werden. Alle anderen Objekte sind Sub-Nodes der Scene.

Die STScene-Klasse beinhaltet momentan keine Vorteile gegenüber der CCScene-Klasse.

4.2.3 STAnimatedSprite
Ein Sprite-Objekt kann ein beliebiges Bild anzeigen. Dies wird in allen 2d Games zur Darstellung verschiedener Spielelemente benutzt.

Ein STAnimatedSprite kann sich hingegen auch noch selber animieren. Hierzu erhält jede Sprite-Klasse ein Plist file, in welchem die gegebenen Animationen definiert sind.

4.2.3.1 STCollidableSprite
Diese Sprite-Klasse wird zur Kollisionen-Erkennung benutzt. Falls eine Kollision auftritt werden die beiden kollidierten Objekte informiert.

4.2.3.1.1 STItemContainer
Ein STItemContainer kann ein oder mehrere Items beinhalten. So wie z.B. viele Münzen, einen Stern, eine Blume, uw.

4.2.3.1.1.1 STBlock
Diese Klasse wird zur Darstellung der Blöcke benutzt, welche Mario zerschlagen kann, um seine Items zu erhalten.

4.2.3.1.2 STCreature
Ein Creature ist ein Spieler-Objekt. Diese haben eine Gesundheit. Falls die Gesundheit 0 oder kleiner ist, sterben die Spieler.

4.2.3.1.2.1 STPlayer
Ein Player ist nun das Objekt, welche der Benutzer steuern kann.

4.2.3.1.2.1.1 STMario
Mario ist nun das definitive Objekt, welches zur Darstellung des Charakteren Mario benutz wird.

4.2.3.1.2.2 STNPC
Diese Klasse wird zum automatischen Bewegen verschiedener Gegner benutzt.

4.2.3.1.2.2.1 STSpiny
Spiny ist ein Gegner mit Stacheln. Mario kann ihn nur durch die Panzer eines Koopas bezwingen.

4.2.3.1.2.2.2 STLakitu
Ein Lakitu ist ein Gegner, welcher auf einer Wolke fliegt.
Er wirft oben genannte Spiny Gegner herunter.

4.2.3.1.2.2.3 STKoopa
Ein Koopa ist ein Gegner mit einem Panzer. Spring man einmal auf ihn, versteckt er sich in seinem Panzer. Den Panzer kann man nun anstossen. Bei einem weiteren Sprung auf den Panzer, stirbt der Koopa.

4.2.3.1.2.2.4 STGumba
Ein Gumba ist ein Gegner, welcher lediglich von links nach rechts spaziert.
Er kann mit einem einfachen Sprung auf den Kopf bezwungen werden. Bei einer Seitwerts-Kollision wird der Spieler beschädigt.

4.2.3.1.3 STItem
Ein Item wird zur einer bestimmten Verbesserung des Players benutzt.
Diese Verbesserungen beziehen sich entweder auf die Punkte-Zahl, auf die Gesundheit, oder anderes.

4.2.3.1.3.1 STCoin
Ein Coin ist eine Münze. Der Spieler kann diese sammeln. Beim Sammeln von je 100 Münzen erhält der Spieler ein Leben.

4.2.3.1.3.2 STMushroom
Ein Mushroom ist ein Pilz. Ein Pilz kann den kleinen Mario zum grossen Mario verwandeln.

4.2.3.1.3.3 STFlower
Eine Flower ist eine Blume. Die Blume kann den kleinen oder grossen Mario zum Feuerblumen-Mario verwandeln.

4.2.3.1.3.4 STBonusItem
Ein Bonus Item ist ein Item, welches nur über eine gewisse Zeit wirkt. Ein Beispiel ist der Stern.

4.2.3.1.3.4.1 STStar
Ein Star ist ein Stern. Ein Stern kann Mario für eine kurze Zeit unverwundbar.

4.2.4 STConfigurationManager
Diese Klasse wird zum Vornehmen der Benutzer-Einstellungen gebraucht.

4.3 Schnittstellen

Wir benutzen Cocos2d als weiteren Layer zwischen Objective-C und OpenGL.
http://www.cocos2d-iphone.org

5 Datenmodell (grob)

Die Levels des Spiels werden in sogenannten Tiled-Maps gespeichert. Das Dateiformat wird von Cocos2d vollumfänglich unterstützt. Durch die sehr gute Unterstützung sind keine Kenntnisse über den Aufbau der TMX-Dateien nötig.

Die Speicherung der Welten und Levels wird in sogenannten Plist (Property Lists) Files abgelegt. Im Key „Naming Convention“ ist das Format des Namens aller Tiled-Maps definiert. So hat jede Welt eine Vielzahl an Levels. Es sind weitere Attribute für die Welten und Levels definiert, welche im Bild zu sehen sind.

Key	Type	Value
▼ Root	Dictionary	(2 items)
Naming Convention	String	world_%d_level_%d.tmx
▼ Worlds	Array	(1 item)
▼ Item 0	Dictionary	(5 items)
ID	Number	0
Name	String	Insert World Name here...
Icon	String	Insert Icon Name here...
isLocked	Boolean	NO
▼ Levels	Array	(1 item)
▼ Item 0	Dictionary	(4 items)
ID	Number	0
Name	String	Insert Level Name here...
Icon Name	String	Insert Icon Name here...
isLocked	Boolean	NO

Abbildung 16: Property-List-Datei der Welten und Level

Die Informationen über die Entwickler sind ebenfalls in Property List Files gespeichert und werden vom Spiel eingelesen.

Key	Type	Value
▼ Root	Dictionary	(2 items)
▼ developers	Array	(2 items)
Item 0	String	Ilija Tovilo
Item 1	String	Lukas Seglias
▼ socialMedia	Array	(2 items)
▼ Item 0	Dictionary	(2 items)
url	String	twitter.com/classicGameRemake
icon	String	twitter.png
▼ Item 1	Dictionary	(2 items)
url	String	facebook.com/classicGameRemake
icon	String	facebook.png

Abbildung 17: Property-List-Datei der Entwickler-Informationen

6 Konsequenzen

Soweit sind keine neuen Risiken erschienen.

7 Antrag

Wir beantragen die Genehmigung des vorliegenden Konzeptberichts und die Freigabe für die Realisierungsphase durch den Auftraggeber.

Ort: _____ Datum: _____ Unterschrift: _____