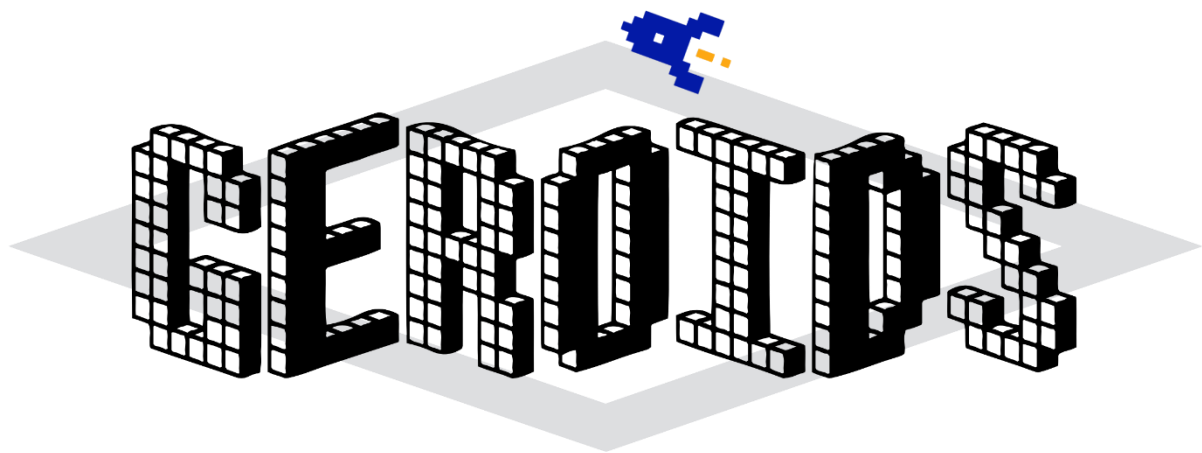




Projektskizze "Geroids"

EIN 2D-ARCADEGAME FÜR KLEIN UND GROSS



Projektteam PSIT3 Gruppe 01

Arben Shabani (PL)
Linda Bödi
Valentin Bossi
Matthias Kaderli



Inhaltsverzeichnis

Inhalt

1. Ausgangslage	3
2. Idee	3
3. Kundennutzen	3
4. Stand der Technik/Konkurrenzanalyse.....	3
5. Hauptanwendungsfall	4
6. Weitere Anforderungen	4
7. Ressourcen	4
8. Risiken.....	5
9. Grobplanung.....	6
9.1 Use-Cases	6
10. Wirtschaftlichkeit	6



1. Ausgangslage

In den frühen 80er Jahren erlebten die Arcade-Spiele einen riesigen Aufschwung, überall wo es Spielhallen gab, konnte man „Asteroids“, „Space Impact“ oder ähnliche Spiele spielen. Während der 90er Jahre nahm dieser Trend ab, um etwa 20 Jahre später mit dem allgemeinen Retro-Hype erneut aufzukommen. Seit ein paar Jahren findet man nun wieder ähnliche Versionen dieser alten Arcade-Spiele, wobei die Nachfrage noch eine ganze Weile anzuhalten scheint.

2. Idee

Geroids ist ein Arcade-Spiel, welches an den Retro-Trend anknüpfen soll. Das Design soll minimalistisch und im Retro-Stil gehalten werden. Es wird Wert auf eine intuitive Spielbedienung gelegt, der Benutzer soll sich nicht ewig in eine Anleitung einlesen müssen. In Geroids steuert der Spieler eine Rakete im Weltall, mit der er geometrischen Figuren ausweichen und/oder diese zerstören muss.

3. Kundennutzen

Geroids macht aufgrund der einfachen Bedienung und des schlichten, schönen Designs grossen Spass. Minimale Spiele sind in einem steigenden Trend, man spricht damit also viele Kunden an. Ganz nach dem Motto *In der Kürze liegt die Würze* ist Geroids auch in einer kurzen Pause spielbar. Braucht ein Mitarbeiter kurzfristig fünf Minuten Pause um sich abzulenken, so ist Geroids eine gute Wahl. Es bietet simple Mechaniken, welche den Kunden unterhalten und seinen Stresslevel reduzieren. Auch für nach dem Feierabend ist Geroids geeignet. Durch ein Level-System mit steigender Schwierigkeit ist auch für Unterhaltung während längerer Zeit gesorgt. Da in dem Spiel auch eine „Werbebläche“ zur Verfügung steht, werden nicht nur Spieler, sondern auch andere Firmen unsere Kunden. Je grösser der Erfolg des Spiels, desto attraktiver wird es für Firmen Werbung zu schalten.

4. Stand der Technik/Konkurrenzanalyse

In der heutigen Zeit bestehen wahnsinnig viele Möglichkeiten Spiele zu designen und zu entwickeln. Die Spiele werden dadurch auch oft immer komplizierter, mit langen Handlungssträngen und vielen Details [1,2]. Dies und Nostalgie sind Gründe dafür, dass es seit mehreren Jahren eine Bewegung zurück zu den Spielen der 80er und 90er Jahre gibt. Das kann man zum Beispiel auch an den hohen Download-Zahlen der Spiele „Flappy Bird“ und „Pokémon Go“ erkennen; Flappy Bird, im Jahr 2013 im Retro-Stil entwickelt, wurde mehr als 50 Millionen Mal heruntergeladen [3] und Pokémon Go, erreichte mit dem Aufgreifen des alten und beliebten Spiels „Pokémon“ eine Vielzahl an Fans, wodurch mehr als 7.5 Millionen Downloads in kürzester Zeit in den USA erreicht wurden [4].

„Asteroids“, „Space Impact“ und „Space Invaders“ gehören zu den ersten und beliebtesten Arcade-Spielen der frühen 80er-Jahre [4]. Mittlerweile sind zwar auch dutzende Versionen dieser Spiele verfügbar, jedoch oft nicht mehr so rudimentär wie früher, sondern mit viel mehr Details. Mit „Geroids“ soll auf den „Retro-Zug“ aufgesprungen werden und ein minimalistisch gehaltenes Spiel entstehen. Die Grafik und das Spiel sollen so simpel sein, dass die Regeln schnell verstanden sind und das Spiel intuitiv gespielt werden kann. Zusätzlich soll es sich dadurch abheben, dass nicht etwa auf Asteroiden oder Meteoriden, sondern auf geometrische Formen geschossen wird.



5. Hauptanwendungsfall

Ein Benutzer möchte das Spiel „Geroids“ spielen und startet die Geroids Applikation auf dem Computer in seinem Webbrowser. Auf dem Bildschirm erscheint die Aufforderung, sich einen Spielernamen zu geben. Der Benutzer bestätigt den gewählten Namen und startet damit das Spiel. Was er nun im Browser sieht, ist die Spielwelt, das Weltall. Mit der Tastatur gibt er Befehle ein, die ein Raumschiff bewegen und schießen lässt. Das Raumschiff repräsentiert den Spieler. Es fliegen geometrische Objekte Richtung Raumschiff. Weil die Objekte das Raumschiff zerstören können, weicht der Spieler ihnen aus oder verteidigt sich mit Schüssen, die die Objekte zerstören. Die Punktzahl des Spielers erhöht sich mit jedem zerstörten Objekt. Nachdem jeweils eine gewisse Punktzahl erreicht ist, erhöht sich die Fluggeschwindigkeit der Objekte, der Schwierigkeitsgrad steigt. Das Spiel ist beendet, sobald das Raumschiff von einem Objekt getroffen und somit zerstört wird. Dem Spieler erscheint eine Rangliste, in der er die erreichte Punktzahl von allen Benutzern dieses Spiels sehen kann. Der Spieler wird bei Ende des Spiels in diese Rangliste entsprechend eingetragen.

6. Weitere Anforderungen

- Der Spielername wird in eine Datenbank gespeichert und ist einmalig.
- Der Benutzer registriert sich mit Name, Email und einem Passwort.
- Die Applikation funktioniert in den aktuellen Versionen von Safari, Firefox, Google Chrome.
- In-App-Käufe ermöglichen das Raumschiff mit zusätzliche Funktionalitäten auszustatten.
- Sicherheit: In-App-Käufe werden mit branchenüblichen Sicherheitsstandards geschützt.
- Mögliche Weiterentwicklungen der Applikation: Multiplayer-Modus, auf Smartphones lauffähig, weitere Schwierigkeitsmodi.

Nicht funktionale Anforderungen:

- OO-Software-Design
- Die Applikation soll von 1'000 Spielern gleichzeitig benutzt werden können.

7. Ressourcen

Die Software-Programmierung benötigt ein gut abgestimmtes Team von 4 Personen. Jedes dieser Teammitglieder muss über ein Know-how in den Bereichen OOP, webbasierte Programmiersprachen und Datenbanken verfügen.

Für die Umsetzung der In-App-Käufe wird externes Wissen benötigt.

Nach den bisherigen Erfahrungen in Projekten wird der Aufwand auf etwa 4 Monate geschätzt, was 640 Arbeitsstunden pro Person entspricht. Somit wird der Gesamtaufwand für die Software auf ungefähr 2500 Arbeitsstunden geschätzt



8. Risiken

Nr.	Name	Beschreibung	Wahrscheinlichkeit	Schaden	Stufe	Priorität	Massnahme
1	Konkurrenz	In dieser Branche herrscht eine grosse Konkurrenz. Schlimmstenfalls geht das Spiel unter, was in kleiner Benutzerzahl resultiert.	50%	Sehr hoch	H	1	Werbung, Mund-zu-Mund-Propaganda
2	Update	Es kann sein, dass das Programm nach einem Browser-Update nicht mehr läuft	30%	Sehr hoch	H	1	Programmierung in Java und JavaScript, wobei beides noch sicher noch weitere Jahre vom Markt unterstützt wird (aufgrund der grossen Verbreitung)
3	Hacker-Attacken	Die Datenbank ist nicht gut gegen konzentrierte Hacker-Attacken geschützt, was eine Gefahr für die Nutzerdaten darstellen kann	5%	Hoch	M	3	Möglicherweise einen Experten hinzuziehen.
4	Design	Es existiert kein grosses Knowhow bezüglich Game-Design	50%	Klein	L	3	Da das Spiel sowieso minimalistisch gehalten wird, sollten dieses fehlende Knowhow keine allzu grosse Gefahr darstellen
5	Spiele im Web	Es gibt kaum Wissen zu Spiel Implementierung im Web	60%	Mittel	M	2	Möglicherweise muss zuerst noch etwas Wissen angeeignet werden, um das Spiel gut implementieren zu können.

Legende: L: Low; M: Medium; H: High



9. Grobplanung

Schätzungsweise 4 Monate wird das gesamte Projekt, wie in Punkt 7 erläutert, dauern. Der Prototyp nimmt gemäss Grobplanung eine Entwicklungszeit von 14 Wochen in Anspruch.

9.1 Use-Cases

Use-Case 1: Registration

Use-Case 2: Spiel starten

Use-Case 3: Spielen

Use-Case 4: Highscore anzeigen

Iteration	Start	Ende	Meilenstein	Aufwand [h]	Ziele / Umsetzung
Inception Phase					
1	20.09.2016	04.10.2016	M1	50	Vision, Projektskizze, Präsentation
Elaboration Phase					
2	04.10.2016	18.10.2016		60	Entwicklungsumgebung, GUI-Entwurf
3	18.10.2016	01.11.2016	M2	80	Domänenmodell, UML-Diagramm, DB-Entwurf
Construction Phase					
4	01.11.2016	15.11.2016		90	SW-Architektur, UC1 & UC2 realisiert
5	15.11.2016	29.11.2016	M3	100	UC3 Implementierung & Testing
6	29.11.2016	06.12.2016		60	UC4 & Datenbank Implementierung
Transition Phase					
7	06.12.2016	20.12.2016	M4	60	UC1-4 GUI-Testing, Prototyp-Release
Total				500	

Für die Entwicklung des Prototyps wird der Gesamtaufwand auf etwa 500 Stunden geschätzt. Innerhalb der 14 Wochen Entwicklungszeit für den Prototyp wird in zweiwöchigen Iterationen gearbeitet.

10. Wirtschaftlichkeit

Wie in Punkt 7 beschrieben beträgt der geschätzte Gesamtaufwand 2500 Arbeitsstunden. Mit einem Jahreslohn von CHF 80'000 pro Person ergeben die Lohnleistungen für das Kernteam zusammen CHF 100'000. Zusätzlich kommen noch geschätzte CHF 50'000 externe Leistungen dazu, was zu einer Gesamtinvestition von ca. 150'000 CHF führt. Wenn unsere Applikation nur ein Prozent der Reichweite des Spieles "Flappy Bird" [6] erreichen kann und somit nur dieses Hundertstel von dessen Tageseinnahmen durch Werbung erreichen kann, also CHF 500 am Tag, wäre die gesamte Investitionssumme innerhalb des ersten Jahres amortisiert. Die restlichen Einnahmen gewonnen durch die In-App-Verkäufe würden die Amortisierungs-Zeit noch weiter verkürzen.



Quellenverzeichnis

- [1] GGS. (30.08.2016). Why Retro Games Are Making a Comeback [Online]. URL: <http://www.ggsgamer.com/2016/08/30/why-retro-games-are-making-a-comeback/> [Stand: 29.9.2016]
- [2] M. Bodmer. (31.08.2016). Schön schräg [Online]. URL: <http://www.nzz.ch/nzzas/nzz-am-sonntag/computergames-schoen-schraeg-ld.113789> [Stand: 29.9.2016]
- [3] A. Freier. (13.04.2015). Flappy Bird: Revenue and Usage Statistics [Online]. URL: <http://www.businessofapps.com/flappy-bird-revenue-and-usage-statistics/> [Stand: 29.9.2016]
- [4] D. Etherington. (11.07.2016). Pokemon Go has an estimated 7.5 M U.S. downloads, £1.6M in daily revenue [Online]. <https://techcrunch.com/2016/07/11/pokemon-go-daily-revenue-downloads/> [Stand: 29.9.2016]
- [5] Goldene Ära der Arcade-Spiele [Online]. URL: https://de.wikipedia.org/wiki/Goldene_Ära_der_Arcade-Spiele
- [6] businessofapps.com (2015). Flappy Bird: Revenue and Usage Statistics [Online]. URL: <http://www.businessofapps.com/flappy-bird-revenue-and-usage-statistics> [Stand: 29.9.2016]