

## **Projekt-Schlussbericht**

**Projekttitle:** PyJump

**Autor(en):** Dominik Schütz  
Raphael Schwob

**Datum:** 06.06.2016

# Abschlussbericht

<b>Status</b>	In Arbeit / <u>In Prüfung</u> / Abgeschlossen
<b>Projektname</b>	PyJump
<b>Projektleiter</b>	Dominik Schütz
<b>Auftraggeber</b>	Daniel Sterchi
<b>Autoren</b>	Dominik Schütz, Raphael Schwob
<b>Verteiler</b>	Daniel Sterchi, Dominik Schütz, Raphael Schwob

## Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
1.0	31.05.16	Erstellung dieses Dokuments	Dominik Schütz
1.1	01.06.16	Weiterführung dieses Dokuments	Dominik Schütz
1.2	05.05.16	Weiterführung dieses Dokuments	Raphael Schwob
2.0	06.06.16	Fertigstellung dieses Dokuments	Dominik Schütz

## Inhaltsverzeichnis

1	Zusammenfassung	6
1.1	Ausgangslage	6
1.2	Vorgehen	6
1.3	Ergebnis	6
1.4	Ausblick	6
2	Zeitplan	7
2.1	Grobplanung der Projektphasen	7
2.2	Effektiver Verlauf der Projektphasen	7
2.3	Planung der Realisierungsphase	8
2.4	Effektiver Verlauf der Realisierungsphase	8
2.5	Planung der Einführungsphase	9
3	Arbeitsprotokoll	10
3.1	Arbeitsprotokoll Raphael Schwob	10
3.2	Arbeitsprotokoll Dominik Schütz	12
4	Situationsanalyse	13
4.1	Ausgangslage	13
4.2	Stärken	13
4.3	Schwächen	13
5	Systemziele	13
5.1	Rahmenbedingungen	14
5.2	Abgrenzung	14
6	Lösungsvarianten	14
6.1	Variantenübersicht	14
6.2	Beschreibung der Variante 1	14
6.3	Beschreibung der Variante 2	15
6.4	Beschreibung der Variante 3	15
6.5	Bewertung der Varianten	16
6.6	Variantenentscheid	16
6.7	Empfehlung	16
7	Systemarchitektur	17
7.1	Systemdesign	17
7.1.1	Struktur	17
7.1.2	Beschreibung der Elemente	17
7.2	Schnittstellendefinitionen	18
7.3	Anforderungszuordnung	18
7.4	Konfigurations-Dokumentation	18
8	Systemtest	19
8.1	Testspezifikation	19
8.1.1	Kritikalität der Funktionseinheit	19
8.1.2	Testanforderungen	19
8.1.3	Testverfahren	20
8.1.4	Testkriterien	20
8.1.5	Testfälle	20
8.2	Testprozedur	22
8.2.1	Vorbereitung	22
8.2.2	Durchführung	22
8.2.3	Nachbearbeitung	22
8.3	Testprotokoll	23
8.3.1	Testobjekt	23
8.3.2	Testresultate	23
8.3.3	Testauswertung	23
8.4	Abnahme	24
9	Einführungskonzept	24
9.1	Einführungsplan	24
9.2	Migrationsplan	25
10	Benutzerdokumentation	25
10.1	Benutzerhandbuch	25
10.1.1	Systemübersicht	25
10.1.2	Anwenderfunktionalität	25
10.2	Supporthandbuch	26

10.2.1	Massnahmen bei Benutzerproblemen .....	26
10.2.2	Massnahmen bei technischen Problemen .....	26
10.2.3	Anhang zum Supporthandbuch.....	27
10.3	Ausbildungsplan .....	27
11	Zielerreichung .....	27
12	Erfahrungen .....	28
12.1	Persönliche Erfahrungen von Dominik Schütz .....	28
12.1.1	Was habe ich gelernt? .....	28
12.1.2	Was verlief gut? .....	29
12.1.3	Was verlief weniger gut? .....	29
12.1.4	Was würde ich in Zukunft anders machen? .....	29
12.2	Persönliche Erfahrungen von Raphael Schwob.....	29
12.2.1	Was habe ich gelernt? .....	30
12.2.2	Was verlief gut? .....	30
12.2.3	Was verlief weniger gut? .....	30
12.2.4	Was würde ich in Zukunft anders machen? .....	30
13	Persönliche Vorsätze für IPA.....	30
13.1	Vorsätze von Dominik Schütz .....	30
13.2	Vorsätze von Raphael Schwob .....	31
14	Glossar.....	32

## Abbildungsverzeichnis

Abbildung 1 Grobplanung Projektphasen.....	7
Abbildung 2 effektiver Verlauf Projektphasen .....	7
Abbildung 3 Planung Realisierungsphase.....	8
Abbildung 4 effektiver Verlauf Realisierungsphase .....	8
Abbildung 5 Planung Einführungsphase .....	9
Abbildung 6 Systemarchitektur.....	17
Abbildung 7 Icon ändern .....	25

## 1 Zusammenfassung

### 1.1 Ausgangslage

Unser Kunde ist ein Fan vom bekannten Smartphone-Spiel Doodle Jump und wünscht sich dieses Spiel auch auf dem MacBook spielen zu können. Leider gibt es keine Version von Doodle Jump für den Mac, welche den Anforderungen des Kunden genügt.

Deshalb wurde in diesem Projekt ein Endlos-Platformer Spiel realisiert, welches Doodle Jump als Vorbild nimmt. Im Fokus stand, dass unser PyJump Spiel einfach und schlank entwickelt wurde und trotzdem folgende Anforderungen erfüllt:

- Ohne Internetverbindung lauffähig
- Werbefrei
- Malwarefrei
- Ohne zusätzliche Software lauffähig

### 1.2 Vorgehen

Wir haben uns dazu entschieden das Spiel in Python zu programmieren, weil Python auf dem Mac bereits vorinstalliert ist und wir in der Python Programmierung bereits Erfahrung haben. Zuerst haben wir die Bewegung der Spielfigur programmiert und danach Funktionen geschrieben um die Spielfigur zu steuern. Nachdem diese beiden Funktionen ausgiebig getestet wurden implementierten wir eine Funktion um Umgebung sprich die Plattformen, über die die Spielfigur springt, zu generieren. Später haben wir die Funktion für die Generierung der Plattformen erweitert, sodass auch Monster generiert werden, die die Spielfigur nicht berühren darf. Als letzten Schritt erweiterten wir die Fähigkeiten der Spielfigur um eine Schiessfunktion, damit die Spielfigur den Monstern entgegenwirken kann.

### 1.3 Ergebnis

Wir konnten unser PyJump Spiel wie in der Zeitplanung vorgesehen rechtzeitig abschliessen. Das Spiel ist lauffähig und enthält alle geforderten Funktionen. Der einzige kleine negative Punkt ist, dass je nach Python Installation noch ein zusätzliches Python Modul nachinstalliert werden muss. Dieses Python Modul ist allerdings schnell mit zwei Befehlen über die Kommandozeile nachinstalliert.

Durch die erfolgreiche Durchführung der Realisierung dieses Projektes konnten wir sehr viel lernen vor allem im Bezug auf die Projektführung und Dokumentation. Zudem konnten wir unsere Python-Kenntnisse verbessern.

### 1.4 Ausblick

Als nächstes gilt es nur noch die Projektpräsentation durchzuführen. Sobald diese durchgeführt ist kann das Projekt erfolgreich abgeschlossen werden.

Da die Realisierungsphase erfolgreich abgeschlossen werden konnte, gibt es keine Nacharbeiten zu erledigen. Probleme oder Risiken bei der Benutzung unseres PyJump-Spiels gibt es auch nicht. Was wir heute allerdings besser machen würden ist das Zeitmanagement respektive die Kommunikation. Durch ein Missverständnis haben wir nämlich den Initialisierungsantrag eine Woche zu spät abgegeben obwohl wir ihn eigentlich fertiggestellt hatten. Dieses Missverständnis bescherte uns einen grossen Punktabzug, aber immerhin kamen wir mit dem Projekt nicht in Rückstand. Im Verlauf der weiteren Projektphasen konnten wir denn Punkteabzug vom Initialisierungsantrag wieder etwas kompensieren, indem wir mehr Zeit und Arbeit investiert hatten.

## 2 Zeitplan

### 2.1 Grobplanung der Projektphasen

		Februar				März				April				Mai				Juni					
Nr.	Projektphase	02.02.16	09.02.16	16.02.16	23.02.16	01.03.16	08.03.16	15.03.16	22.03.16	29.03.16	05.04.16	12.04.16	19.04.16	26.04.16	03.05.16	10.05.16	17.05.16	24.05.16	31.05.16	07.06.16	14.06.16	21.06.16	28.06.16
1	Projektinitialisierungsphase																						
4	Konzeptphase																						
5	Realisierungsphase																						
6	Einführungsphase																						
7	Dokumentationsphase																						
8	Präsentation																						

Abbildung 1 Grobplanung Projektphasen

### 2.2 Effektiver Verlauf der Projektphasen

Nr.	Projektphase	Februar				März				April				Mai				Juni					
		02.02.16	09.02.16	16.02.16	23.02.16	01.03.16	08.03.16	15.03.16	22.03.16	29.03.16	05.04.16	12.04.16	19.04.16	26.04.16	03.05.16	10.05.16	17.05.16	24.05.16	31.05.16	07.06.16	14.06.16	21.06.16	28.06.16
1	Projektinitialisierungsphase																						
4	Konzeptphase																						
5	Realisierungsphase																						
6	Einführungsphase																						
7	Dokumentationsphase																						
8	Präsentation																						

Abbildung 2 effektiver Verlauf Projektphasen

Wie man sieht wurde die Realisierungsphase vom Auftraggeber verlängert. Dies kam uns gelegen, sodass wir mehr Zeit in die Entwicklung unseres PyJump-Spiel investieren konnten.

## 2.3 Planung der Realisierungsphase

Nr.	Projektphase	April						
		22.03.16	29.03.16	05.04.16	12.04.16	19.04.16	26.04.16	
		1	main.py					
		2	plattform.py					
		3	player.py					
		4	monster.py					
		5	Abschlusstests					

Abbildung 3 Planung Realisierungsphase

## 2.4 Effektiver Verlauf der Realisierungsphase

Nr.	Projektphase	März		April				Mai	
		22.03.16	29.03.16	05.04.16	12.04.16	19.04.16	26.04.16	03.05.16	10.05.16
1	main.py								
2	plattform.py								
3	player.py								
4	monster.py								
5	Abschlusstests								

Abbildung 4 effektiver Verlauf Realisierungsphase

Wie in der Grobplanung bereits erwähnt, wurde die Realisierungsphase um zwei Wochen verlängert. Wir investierten diese Zeit in die drei wichtigsten Module: main.py, plattform.py und player.py.



## 2.5 Planung der Einführungsphase

		Mai			
Nr.	Projektphase	17.05.16		24.05.16	
1	Spiel installieren				
2	Kunde instruieren				
3	Abnahme durchführen				
4	Abschluss vorbereiten				

Abbildung 5 Planung Einführungsphase

Während der Einführungsphase gab es keine Probleme, sodass diese wie geplant durchgeführt wurde.

## 3 Arbeitsprotokoll

### 3.1 Arbeitsprotokoll Raphael Schwob

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Grundstruktur für Code erstellen	1	0.5
Grundlegende Funktionen implementieren	1	1
<b>Tages Ablauf</b>		
<p>Da ich heute die ersten Zeilen Code geschrieben habe, musste ich natürlich als erstes eine Grundstruktur aufbauen, auf der ich dann in den kommenden Wochen den Rest des Spieles aufbauen kann. Wir haben komplett ohne Framework gearbeitet, daher gab es noch keinen vorgegebenen Code, und ich musste ganz bei 0 anfangen. Allerdings habe ich den Aufwand dann doch etwas überschätzt. Bis ich ein ausführbares Python Programm hatte, habe ich nur 30min gebraucht, und nicht wie vermutet 1 Stunde.</p> <p>Danach habe ich auch schon anfangen können, die wichtigsten Funktionalitäten zu implementieren. Das Wichtigste war für mich die Fähigkeit der Spielerfigur, sich nach oben fortbewegen zu können. Deshalb habe ich auch direkt damit angefangen. Am Ende ist die Spielfigur automatisch nach oben gesprungen und der Hintergrund hat sich passend dazu nach unten bewegt.</p>		
<b>Hilfestellungen</b>		
<p>Da ich vor einigen Wochen als Einstieg an meinem neuen Arbeitsplatz bereits ein kleines Python Game mit dem tkinter machen konnte, habe ich mich natürlich auch ein wenig am Code orientiert, den ich bei diesem Projekt geschrieben habe. Vor allem die Grundstruktur konnte ich recht gut wiederverwenden, da diese sich kaum geändert hat.</p>		
<b>Reflexion</b>		
<p>Ich konnte so ziemlich alle grundlegenden Dinge bereits implementieren und habe nun eine gute Basis, auf der ich nun das weitere Projekt aufbauen kann. Insgesamt habe ich etwas weniger Zeit gebraucht als erwartet, allerdings konnte ich dadurch auch ganz einfach noch ein wenig mehr an den Funktionalitäten arbeiten.</p>		
<b>Nächste Schritte</b>		
<p>Mit dem nun mehr oder weniger fertigen Grundgerüst, kann ich nun anfangen die verschiedenen Funktionen die noch benötigt werden einzubauen. Als nächstes werde ich versuchen das Game spielbar zu machen, in dem ich die Plattformen zum darauf herumspringen einbauen werde. Wenn diese erst einmal funktionieren, ist der Rest nur noch etwas Feintuning.</p>		

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Game spielbar machen	1	1
Feintuning	0.5	1
<b>Tages Ablauf</b>		
<p>Als erste Arbeit stand heute das Implementieren weiterer Funktionen und Features an, um das Game nun spielbar zu machen. Als erstes habe ich also mit dem etwa wichtigsten Feature angefangen, den Plattformen auf denen der Spieler springen kann. Nachdem diese nach ca. 30min erfolgreich im Spiel eingebaut waren, habe ich in den verbleibenden 30min noch weitere Grundfunktionen wie zum Beispiel die Punkteanzeige eingebaut.</p> <p>Nachdem das Spiel nun spielbar war, musste ich noch verschiedene Parameter die einen Einfluss auf das Spielerlebnis des Spielers haben einstellen. Dazu gehörten Dinge wie zum Beispiel die Spawnhäufigkeit von Monstern oder die Sprunghöhe des Spielers. Das Wichtigste dabei war natürlich die Punkteverteilung. Da die Scores am Ende jedes Spiels abgespeichert werden, habe ich versucht die Punkte so fair wie möglich zu verteilen.</p>		
<b>Hilfestellungen</b>		
<p>Als Hilfestellung habe ich auch heute wieder den Code von meinem vorherigen Projekt verwendet. Auch wenn ich dort ein komplett anderes Spiel entwickelt habe, konnte ich doch recht viele der Techniken die ich dort verwendet habe auch in diesem Projekt wiederverwenden. Zusätzlich dazu habe ich diesmal auch noch ein wenig die tkinter Doku verwendet. In der findet man so ziemlich alles was man irgendwie gebrauchen könnte in der Entwicklung mit dem Python Paket.</p>		
<b>Reflexion</b>		
<p>Ich bin recht gut vorwärtsgekommen und konnte so ziemlich alle Features implementieren. Das Spiel ist nun spielbar und recht gut ausbalanciert. Einige Dinge waren recht schnell gemacht, da sie teilweise eins zu eins mit meinem vorherigen Projekt übereinstimmten. Andere Dinge wiederum wie zum Beispiel das Spawnen der Monster hat etwas länger gedauert, da es etwas komplizierter ist und ich noch keine Erfahrung damit hatte.</p>		
<b>Nächste Schritte</b>		
<p>Als nächstes werde ich nun nochmal die Doku durchlesen und nach Features suchen, die ich noch nicht implementiert habe. Danach sollte ich immer noch etwas Zeit haben, gewisse Dinge zu optimieren. Soweit ich das aber momentan im Kopf habe, habe ich inzwischen so ziemlich alles, was das Spiel benötigt bereits eingebaut.</p>		

## 3.2 Arbeitsprotokoll Dominik Schütz

Tätigkeiten	Aufwand geplant (Std)	Aufwand effektiv (Std)
Realisierungsbericht schreiben	3	4
Spiel testen	0.5	0.5
<b>Tages Ablauf</b>		
<p>Heute habe ich am Realisierungsbericht gearbeitet. Parallel dazu hat Raphael an unserem PyJump-Spiel entwickelt. Als erstes habe ich mich um die Teile gekümmert, welche wir im Konzeptbericht teilweise bereits geschrieben haben. Diese habe ich übernommen und wo nötig angepasst, damit ich sie im Realisierungsbericht passend integrieren kann.</p>		
<p>Nun wo ich die ersten Teile des Realisierungsberichts geschrieben habe schaute ich mal wie weit Raphael mit dem Spiel ist. Ich schaute mir den Code an und fing an diesen in der Systemdokumentation zu dokumentieren. Zuerst schrieb ich eine kurze Einleitung, danach suchte ich nach wichtigen Variablen und beschrieb diese in der Konfigurations-Dokumentation.</p>		
<p>Als ich die Systemdokumentation fertig hatte fing ich mit dem Benutzerhandbuch an. Auch hier schrieb ich zuerst eine kurze Einleitung, danach beschrieb ich die Installation und Bedienung des Spiels. Für schlechte Zeiten habe ich ebenfalls die Problembehandlung und Wiederherstellung beschrieben.</p>		
<p>Nach dem Benutzerhandbuch habe ich mich um das Supporthandbuch gekümmert. Dies gab nicht sehr viel zu tun, da wir nicht viel Supporten können. Falls es beim Spielen Probleme geben sollte kann man nicht viel Anderes machen, als das Spiel neu zu starten.</p>		
<p>Im Supporthandbuch habe ich auch beschrieben, was zu tun ist bei technischen Problemen. Zum Beispiel wie die Python Image Library Pillow mit <code>easy_install</code> und <code>pip</code> über den Terminal nachinstalliert werden kann.</p>		
<p>Nachdem der theoretische Teil des Realisierungsberichts fertig war fing ich mit dem Testen unseres PyJump-Spiel an. Im Konzeptbericht haben wir bereits ein paar Testfälle definiert. Diese habe ich übernommen und ausgearbeitet. Zudem habe ich eine Testspezifikation geschrieben um zu definieren, wie, mit welchen Vorbereitungen, Nachbereitungen und nach welchen Kriterien getestet werden soll. Danach führte ich die Testfälle aus. Dies beanspruchte nicht sehr viel Zeit, weil alles auf anhielt funktionierte.</p>		
<b>Hilfestellungen</b>		
<p>Einige Teile konnte ich vom Konzeptbericht übernehmen. Zudem half mir die Vorlage, welche wir erhielten sehr den Realisierungsbericht zu strukturieren.</p>		
<b>Reflexion</b>		
<p>Beim Realisierungsbericht bin ich grösstenteils mit meiner Arbeit zufrieden. Allerdings hätte ich wahrscheinlich die Schnittstellen des Spiels besser beschreiben können, wie diese funktionieren und welche Abhängigkeiten darin bestehen.</p>		
<b>Nächste Schritte</b>		
<p>Als nächstes kriegt der Realisierungsbericht den letzten feinschliff und danach starten wir in die Einführungsphase.</p>		

## 4 Situationsanalyse

### 4.1 Ausgangslage

Auf dem MacBook unseres Kunden ist derzeit kein einziges Spiel installiert. Unser Kunde ist mit dem Angebot des App Stores nicht zufrieden. Er wünscht sich ein ressourcenfreundliches Spiel, dass auch für seine Kinder geeignet ist. Da unser Kunde ein Fan von Plattformern ist, wünscht er sich ein solches Endlosspiel auf seinem MacBook.

Als Vorbild sieht er das Smartphone-Spiel Doodle Jump. Unser Kunde hat bereits im Internet nach einer Doodle Jump Version für den Mac gesucht, konnte aber keine Version finden die seinen Ansprüchen genügen. Er fand eine Doodle Jump online Version, allerdings ist das auch keine Lösung, da sein Spiel auch offline verfügbar sein muss. Die bei der Internetrecherche gefundenen offline Versionen von Doodle Jump haben unseren Kunden auch nicht überzeugt, weil diese aus nicht transparenten Quellen stammen.

Nach der erfolgreichen Durchführung dieses Projekts hat der Kunde einen lang ersehnten, endlosen Platformer auf seinem MacBook, den er auch ohne Internetzugang verwenden kann.

### 4.2 Stärken

Derzeit sind vergleichbare Online-Spiele verfügbar, welche lauffähig sind ohne zusätzliche Software zu installieren. Die vergleichbaren Online-Spiele sind auf jedem Computer mit einem gängigen Browser lauffähig, sofern eine Internetverbindung besteht.

Unser Produkt wird diese Stärken mitnehmen. Es wird auf dem Zielgerät(MacBook) lauffähig sein ohne zusätzliche Software zu installieren. Auf allfälligen zusätzlichen Geräten muss Python installiert werden. Bei vielen Unix ähnlichen Betriebssystemen ist dies bereits bei der Grundinstallation der Fall. Zudem wird unser Produkt auch ohne Internetverbindung lauffähig sein.

### 4.3 Schwächen

Schwachpunkt	Beschreibung	Verbesserung
S1	Es existiert keine legitime Quelle für ein vergleichbares Spiel.	Unser Spiel ist von vertrauenswürdigen Entwicklern programmiert und frei von Malware.
S2	Es bestehen vergleichbare Online-Spiele.	Unser Spiel ist offline verfügbar.
S3	Die vergleichbaren Online-Spiele enthalten Werbung	Unser Spiel ist Werbefrei.
S4	Bei Online-Spielen besteht die Gefahr, dass der Server ausfällt und das Spiel somit nicht verfügbar ist.	Unser Spiel läuft lokal und ist somit unabhängig von einem Server.

## 5 Systemziele

Ziel	Beschreibung	Schwachpunkte
Z1	Das Spiel ist ohne Malware.	S1
Z2	Das Spiel ist offline verfügbar.	S2, S4
Z3	Das Spiel ist ohne zusätzlich installierte Software lauffähig.	S2, S4
Z4	Das Spiel läuft endlos weiter bis der Spieler einen Fehler macht.	
Z5	Dem Spieler wird der Punktestand laufend angezeigt.	
Z6	Das Spiel ist werbefrei.	S3

## 5.1 Rahmenbedingungen

- Organisatorische Rahmenbedingungen
  - Anwendung der HERMES gibb Projektmethode
- Finanzielle Rahmenbedingungen
  - MacBook mit installierten Office-Anwendungen bereits vorhanden
  - MacBook mit installiertem Python bereits vorhanden
- Zeitliche Rahmenbedingungen
  - Projektinitialisierungsauftrag erteilt (09.02.2016)
  - Initialisierungsphase abgeschlossen (01.03.2016)
  - Konzeptphase abgeschlossen (15.03.2015)
  - Realisierungsphase abgeschlossen (26.04.2016)
  - Einführungsphase abgeschlossen (10.05.2016)
  - Schlussbericht fertig gestellt (24.05.2016)
  - Präsentation (24.05.2016 – 31.05.2016)
- Räumliche Rahmenbedingungen
  - Zimmer IE103

## 5.2 Abgrenzung

- Es wird keine neue Hardware angeschafft.
- Auf Hardware und Betriebssystem wird kein Support geleistet.
- Das Spiel wird ausschliesslich mit Python 2.7 entwickelt und getestet.
- Es wird nicht garantiert, dass unser PyJump Spiel auch mit anderen Python-Versionen lauffähig ist.

## 6 Lösungsvarianten

### 6.1 Variantenübersicht

- Variante 1
  - Das Spiel wird als Webapplikation realisiert und wird über einen Browser gespielt.
- Variante 2
  - Das Spiel wird mit Python realisiert.
- Variante 3
  - Das Spiel wird mit Java realisiert.

### 6.2 Beschreibung der Variante 1

Bei der Variante 1 wird das Spiel als Webapplikation mittels JavaScript und PHP realisiert. Mit dieser Variante muss das Spiel nicht installiert werden. Man kann das Spiel einfach auf jedem Computer mit Internetverbindung und einem gängigen Browser spielen.

Bei dieser Variante wird zusätzlich noch ein Apache-Webserver benötigt, von dem aus das Spiel bereitgestellt wird. Als Betriebssystem für den Webserver wird ein Red Hat Enterprise Linux 7 verwendet. Auf dem Webserver wird PHP installiert.

Das Gameplay wird sich daraus auszeichnen, dass der Spieler die Spielfigur durch springen über mehrere Plattformen nach oben bringen muss. Der Spieler erhält dann je nach überwandener Höhe mehr Punkte. Sobald der Spieler eine Plattform verfehlt und hinunterfällt ist das Spiel vorbei. Während des Spiels wird dem Spieler die aktuelle Punktzahl laufend angezeigt.

Kosten für Sachaufwände:

Anzahl	Artikel	Preis in CHF
1	Hardwarekosten für Webserver	1600
1	Red Hat Standard Subscription	800
<b>Total</b>		<b>2400</b>

Kosten für Personalaufwände:

Stunden	Dienstleistung	Preis in CHF
4	Inbetriebnahme des Webserver	400
40	Programmierung des Spiels	4000
<b>Total</b>		<b>4400</b>

### 6.3 Beschreibung der Variante 2

Bei der Variante 2 wird das Spiel mittels Python realisiert. Das Spiel wird auf dem Computer installiert und kann dann lokal gespielt werden. Mit dieser Variante wird lokal mehr Speicherplatz benötigt als mit Variante 1, dafür ist das Spiel auch Offline verfügbar. Da auf dem Zielgerät(MacBook) bereits Python installiert ist, muss keine zusätzliche Software installiert werden.

Das Gameplay wird sich daraus auszeichnen, dass der Spieler die Spielfigur durch springen über mehrere Plattformen nach oben bringen muss. Der Spieler erhält dann je nach überwundener Höhe mehr Punkte. Sobald der Spieler eine Plattform verfehlt und hinunterfällt ist das Spiel vorbei. Während des Spiels wird dem Spieler die aktuelle Punktzahl laufend angezeigt.

Kosten für Personalaufwände:

Stunden	Dienstleistung	Preis in CHF
4	Inbetriebnahme des Webserver	400
40	Programmierung des Spiels	4000
<b>Total</b>		<b>4400</b>

### 6.4 Beschreibung der Variante 3

Bei der Variante 3 wird das Spiel mittels Java realisiert. Das Spiel wird auf dem Computer installiert und kann dann lokal gespielt werden. Mit dieser Variante wird lokal mehr Speicherplatz benötigt als mit Variante 1, dafür ist das Spiel auch Offline verfügbar.

Damit das Spiel mit dieser Variante realisiert werden kann, muss auf dem Zielgerät(MacBook) zusätzlich Java installiert werden.

Das Gameplay wird sich daraus auszeichnen, dass der Spieler die Spielfigur durch springen über mehrere Plattformen nach oben bringen muss. Der Spieler erhält dann je nach überwundener Höhe mehr Punkte. Sobald der Spieler eine Plattform verfehlt und hinunterfällt ist das Spiel vorbei. Während des Spiels wird dem Spieler die aktuelle Punktzahl laufend angezeigt.

Kosten für Personalaufwände:

Stunden	Dienstleistung	Preis in CHF
4	Inbetriebnahme des Webserver	400
40	Programmierung des Spiels	4000
<b>Total</b>		<b>4400</b>

## 6.5 Bewertung der Varianten

Anforderung	1	2	2	Bemerkung
Das Spiel ist Offline Verfügbar.	x	✓	✓	
Das Spiel ist ohne zusätzliche Software lauffähig.	x	✓	x	Python ist bereits installiert.
Es muss keine zusätzliche Hardware angeschafft werden	x	✓	✓	MacBook ist bereits vorhanden.
Das Spiel muss nicht installiert werden.	✓	x	x	
Das Spiel läuft endlos.	✓	✓	✓	
Der Punktestand wird fortlaufend angezeigt.	✓	✓	✓	
Verfehlt die Spielfigur eine Plattform so wird das Spiel beendet.	✓	✓	✓	
Das Spiel enthält keine Malware.	✓ x	✓	✓	Bei Variante 1 besteht die Möglichkeit, dass sich über den Webserver Malware einschleichen könnte.

## 6.6 Variantenentscheid

Als Lösung dient die Variante 2, weil mit ihr, wie bei Abschnitt 6 dargestellt wird, alle Muss-Anforderungen erfüllt werden. Somit wird das Spiel mit Python realisiert. Dadurch ist es Offline-Verfügbar und es wird keine zusätzliche Software benötigt, weil Python bereits installiert ist.

Das Gameplay wird sich daraus auszeichnen, dass der Spieler die Spielfigur durch springen über mehrere Plattformen nach oben bringen muss. Der Spieler erhält dann je nach überwundener Höhe mehr Punkte. Sobald der Spieler eine Plattform verfehlt und hinunterfällt ist das Spiel vorbei. Während des Spiels wird dem Spieler die aktuelle Punktzahl laufend angezeigt.

Anforderung	Beschreibung	Ergebnis
A1	Das Spiel darf keine Malware enthalten.	✓
A2	Das Spiel muss offline gespielt werden können.	✓
A3	Das Spiel muss ohne zusätzlich installierte Software lauffähig sein.	✓
A4	Das Spiel darf an keine Höchstpunktzahl gebunden sein.	✓
A5	Der Punktestand muss laufend aktualisiert werden.	✓
A6	Das Spiel muss werbefrei sein.	✓

## 6.7 Empfehlung

Wir empfehlen die Lösungsvariante 2 zu wählen, weil diese Variante bei Abschnitt 6, der Variantenbewertung klar hervorsticht. Zudem erfüllt sie wie bei der Lösungsbeschreibung in Abschnitt 7 gezeigt alle Anforderungen.

Mit Variante 2 ist es möglich das Spiel ohne Sicherheitsbedenken und ohne zusätzlich installierter Software offline zu spielen.



## 7 Systemarchitektur

### 7.1 Systemdesign

#### 7.1.1 Struktur

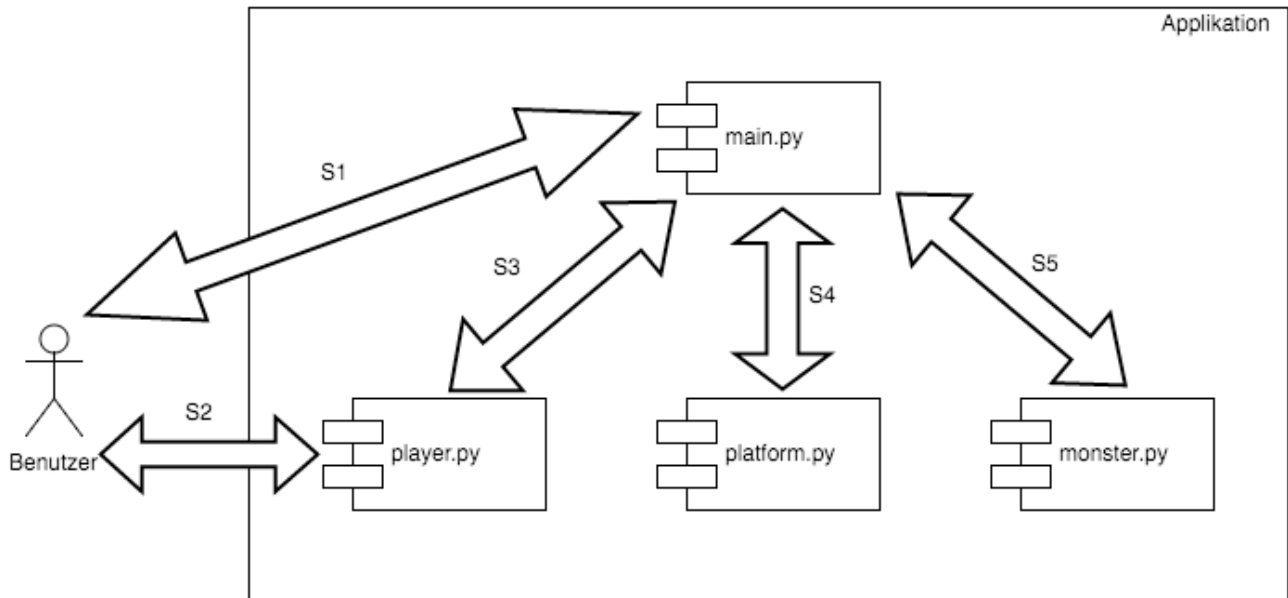


Abbildung 6 Systemarchitektur

#### 7.1.2 Beschreibung der Elemente

Modul	Beschreibung	Verwendung
main.py	Main File des Games	Steuert die Logik sowie alle Prozesse rund um den Inhalt des Spiels. Zudem werden alle Subklassen von hier aus kontrolliert.
platform.py	Enthält die Klasse für die Plattformen	Die Klasse wird als Grundlage für alle Variationen von Plattformen die im Spiel vorkommen können verwendet. Die Bewegungen sowie spezielle Attribute der Plattformen werden hier kontrolliert.
player.py	Enthält die Klasse für eine Spielfigur die vom Spieler gesteuert werden kann	In dieser Klasse werden die Bewegungen des Spielers aufgenommen und verarbeitet. Dabei werden auch die verschiedenen Useringaben verarbeitet.
monster.py	Hauptklasse für verschiedene Typen von Monstern	Die möglichen Variationen von Monstern die im Spiel vorkommen können verwenden alle diese Klasse als Grundlage. Die Klasse steuert das Verhalten und vor allem die Bewegungen der Monster.

## 7.2 Schnittstellendefinitionen

### Externe Schnittstellen

Als einzige externe Schnittstelle dient das GUI vom Game. Durch die simple Bedienung per Tastatur kann der Spieler das Spiel steuern. Zudem gibt es vor Start eines Games immer einen kleinen Dialog in dem ein paar Optionen angepasst werden können. Durch das GUI, welches jeweils vor Spielbeginn aufgerufen wird, kann der Spieler alle nötigen Interaktionen mit dem Spiel tätigen. Dieses GUI wird auch im Pausenmenu angezeigt, welches mit der Taste „P“ aufgerufen wird.

Schnittstelle	Beschreibung
S1	Diese Schnittstelle dient der Verbindung zwischen dem Benutzer und dem Menu. Über diese Schnittstelle kann der Benutzer die Spieloptionen einstellen.
S2	Mit dieser Schnittstelle wird die Spielfigur vom Modul player.py durch die Tastatureingaben vom Benutzer gesteuert.

### Interne Schnittstellen

Als interne Schnittstelle dient, wie in Punkt 3 bereits beschrieben, die Main Klasse des Spiels. Über diese Klasse können die einzelnen Module untereinander und mit der Hauptklasse selbst kommunizieren. Die Hauptklasse dient dabei als Koordinator der ganzen Kommunikation, sie delegiert, wenn nötig gewisse Aufgaben an die einzelnen Module. Zudem basiert das Spielfeld auf der Mainklasse, wodurch sie auch verantwortlich für das ganze Geschehen auf dem Spielfeld ist.

Schnittstelle	Beschreibung
S3	Das Modul main.py steuert über diese Schnittstelle das Modul player.py.
S4	Das Modul main.py steuert über diese Schnittstelle das Modul platform.py.
S5	Das Modul main.py steuert über diese Schnittstelle das Modul monster.py.

## 7.3 Anforderungszuordnung

AFo.-Nr	Anforderung (Stichwort)	main.py	platform.py	player.py	monster.py
A1	Malwarefrei	x	x	x	x
A2	Offline verfügbar	x	x	x	x
A3	Installationsfrei	x	x	x	x
A4	Endlosspiel	x			
A5	Punkte aktualisieren	x			
A6	Werbefrei	x	x	x	x

## 7.4 Konfigurations-Dokumentation

Bei diesem Projekt wird ein Platformer-Spiel umgesetzt, welches Offline genutzt werden kann. Somit ist eine besondere Netzwerkkonfiguration nicht nötig. Bei diesem Spiel gibt es auch keine anderen Konfigurationsdateien. Das Verhalten des Spiels wird durch ein paar wichtige Variablen gesteuert. Diese wichtigen Variablen werden nachführend aufgelistet und erklärt.

## Globale Variablen

Variable	Bemerkung	Standardwert
WIDTH	Breite des Fensters	500
HEIGHT	Höhe des Fensters	750
DELAY	Dauer des Timers	50
ROOT_DIR	Wurzelverzeichnis	os.path.abspath(os.path.join(__file__, "..", ".."))
VERSION	Aktuelle Version des Spiels	1.0.0

## Lokale Variablen von initGame

Variable	Bemerkung	Standardwert
gameOver	Boolean bestimmt ob das Spiel noch läuft	False
gamePaused	Boolean bestimmt ob das Spiel pausiert ist	False
score	Punktestand des Spielers	0

## Lokale Variablen von initObj

Variable	Bemerkung	Standardwert
bg_img	Definiert das Hintergrundbild	ImageTk.PhotoImage(file=ROOT_DIR + "/pyjump/gfx/bg.png")
player_img	Definiert die Spielfigur	ImageTk.PhotoImage(file=ROOT_DIR + "/pyjump/gfx/rocket.png")
monster_img	Definiert das Monster	ImageTk.PhotoImage(file=ROOT_DIR + "/pyjump/gfx/monster.png")
topBarFont	Definiert Schriftgrösse	tkFont.Font(size="20")

## 8 Systemtest

### 8.1 Testspezifikation

#### 8.1.1 Kritikalität der Funktionseinheit

Kritikalität	Beschreibung
Erfüllt	Test wurde erfolgreich abgeschlossen. Keine Nacharbeiten nötig
Teilweise erfüllt	Test wurde teilweise erfolgreich abgeschlossen. Kleine Nacharbeiten nötig.
Nicht erfüllt	Test wurde nicht erfolgreich abgeschlossen. Grössere Nacharbeiten nötig.

#### 8.1.2 Testanforderungen

Die Tests werden unter normalen möglichst betriebsnahen Bedingungen durchgeführt, sodass wir möglichst realitätstreue Ergebnisse erhalten. Bei den Teilen wo mit Daten gearbeitet wird, werden wir zusätzlich noch mit ungültigen Werten testen.

Die Tests unter Höchstleistung oder mit einem Komponentenausfall durchzuführen ist nicht nötig, da unser Spiel nicht auf Redundanz oder Hochverfügbarkeit ausgelegt ist.

### 8.1.3 Testverfahren

- Vorbereitung: Die Software wird auf das Gerät des Kunden kopiert und dann normal gestartet.
- Durchführung: Die Tests werden alle gemäss den Testfällen auf der vorbereiteten Umgebung durchgeführt.
- Auswertung: Die Tests werden gemäss den Testfällen ausgewertet und im Testprotokoll festgehalten. Die Bewertung der Tests findet gemäss der Kritikalität statt, somit müssen Tests, welche nicht erfüllt wurden behoben werden.

### 8.1.4 Testkriterien

- Abdeckungsgrad: Es wird so viel getestet, dass garantiert werden kann, dass alle Anforderungen durch das Spiel abgedeckt sind und funktionieren.
- Checklisten: Als Checklisten dienen die Testfälle.
- Ende-Kriterien: Ein Test gilt als erfolgreich abgeschlossen, wenn er gemäss Kritikalität als „erfüllt“ bewertet wird.

### 8.1.5 Testfälle

Nr.	AFo-Nr.	Anwendungsfall (ggf. orientiert an Use Cases)	Ausgangssituation	Eingabedaten	erwartetes Ergebnis	Bemerkungen, Prüfergebnis
T1	A2	Spiel ist lauffähig ohne Internetverbindung	Das Spiel wird gestartet.		Das Spiel startet und zeigt das Startmenü an.	Erfüllt
T2	A2	Spiel ist lauffähig ohne Internetverbindung	Der Tester drückt im Startmenü die Leertaste und das Spiel beginnt.		Das Spiel startet und läuft bis der Tester einen Fehler macht oder das Spiel abbricht.	Erfüllt
T3	A2	Spiel ist lauffähig ohne Internetverbindung	Das Spiel wurde gestartet.	Der Tester versucht die Spielfigur durch drücken der Taste „A“ nach links zu bewegen.	Die Spielfigur bewegt sich nach links.	Erfüllt
T4	A2	Spiel ist lauffähig ohne Internetverbindung	Das Spiel wurde gestartet	Der Tester versucht die Spielfigur durch drücken der Taste „D“ nach rechts zu bewegen.	Die Spielfigur bewegt sich nach rechts.	Erfüllt
T5	A2	Spiel ist lauffähig ohne Internetverbindung	Das Spiel wurde gestartet	Der Tester drückt die Taste „P“ um	Das Spiel pausiert.	Erfüllt

				das Pausenmenu aufzurufen.		
T6	A4.1 A4.2	Das Spiel ist an keine Höchstpunktzahl gebunden.	Das Spiel wurde gestartet.		Das Spiel läuft weiter bis der Tester einen Fehler macht.	Erfüllt
T7	A5.1 A5.2	Der Punktestand wird aktualisiert und angezeigt.	Das Spiel wurde gestartet.		Der Punktestand wird oben rechts angezeigt und laufend aktualisiert.	Erfüllt
T8	A6	Das Spiel ist werbefrei.	Das Spiel wurde gestartet.		Während des Spielens und im Startmenü wird keine Werbung angezeigt.	Erfüllt

## 8.2 Testprozedur

### 8.2.1 Vorbereitung

Nr.	Vorbereitung
T1	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li></ul>
T2	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li></ul>
T3	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li><li>• Der Tester beginnt das Spiel durch drücken der „Leertaste“</li></ul>
T4	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li><li>• Der Tester beginnt das Spiel durch drücken der „Leertaste“</li></ul>
T5	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li><li>• Der Tester beginnt das Spiel durch drücken der „Leertaste“</li></ul>
T6	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li><li>• Der Tester beginnt das Spiel durch drücken der „Leertaste“</li></ul>
T7	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li><li>• Der Tester beginnt das Spiel durch drücken der „Leertaste“</li></ul>
T8	<ul style="list-style-type: none"><li>• Spiel ist auf dem Testgerät installiert</li><li>• Netzwerkverbindung ist getrennt</li><li>• Der Tester beginnt das Spiel durch drücken der „Leertaste“</li></ul>

Voraussetzungen:

- Python 2.7 muss installiert sein
- Pillow Modul muss installiert sein
- Je nach Installationsart muss eine Internetverbindung vorhanden sein

Konfiguration:

Es wird keine besondere Konfiguration benötigt. Es muss lediglich das Spiel auf das Testgerät kopiert werden. Falls Pillow noch nicht installiert ist kann dies wie folgt über den Terminal erledigt werden:

```
sudo easy_install pip
sudo pip install pillow
```

### 8.2.2 Durchführung

Die Testdurchführung und der Ablauf werden in den jeweiligen Testfällen ausführlich beschrieben.

### 8.2.3 Nachbearbeitung

Die Resultate sollen unter dem Abschnitt 4.3.2 Testresultate dokumentiert werden. Im Abschnitt 4.3.3 Testauswertung soll dann ein Fazit aus den Testresultaten gezogen werden.

### 8.3 Testprotokoll

#### 8.3.1 Testobjekt

Nr.	Tester	Ort	Datum	Zeit	Bemerkungen
T1	Dominik Schütz	Grosshöchstetten	10.05.16	20:45	Testfall erfüllt
T2	Dominik Schütz	Grosshöchstetten	10.05.16	20:48	Testfall erfüllt
T3	Dominik Schütz	Grosshöchstetten	10.05.16	20:50	Testfall erfüllt
T4	Dominik Schütz	Grosshöchstetten	10.05.16	20:52	Testfall erfüllt
T5	Dominik Schütz	Grosshöchstetten	10.05.16	20:55	Testfall erfüllt
T6	Dominik Schütz	Grosshöchstetten	10.05.16	20:58	Testfall erfüllt
T7	Dominik Schütz	Grosshöchstetten	10.05.16	21:01	Testfall erfüllt
T8	Dominik Schütz	Grosshöchstetten	10.05.16	20:03	Testfall erfüllt

#### 8.3.2 Testresultate

Nr.	Testergebnis	Erwartetes Ergebnis	Bemerkungen
T1	Das Spiel ist gestartet und das Startmenü wird angezeigt	Das Spiel startet und zeigt das Startmenü an.	Testfall erfüllt
T2	Das Spiel läuft bis ich einen Fehler mache.	Das Spiel startet und läuft bis der Tester einen Fehler macht oder das Spiel abbricht.	Testfall erfüllt
T3	Die Spielfigur bewegte sich nach links.	Die Spielfigur bewegt sich nach links.	Testfall erfüllt
T4	Die Spielfigur bewegte sich nach rechts.	Die Spielfigur bewegt sich nach rechts.	Testfall erfüllt
T5	Das Spiel hält an und es wird ein Label mit der Aufschrift „Pause“ angezeigt.	Das Spiel pausiert.	Testfall erfüllt
T6	Das Spiel stoppt sobald ich einen Fehler mache.	Das Spiel läuft weiter bis der Tester einen Fehler macht.	Testfall erfüllt
T7	Der Punktestand wird laufend aktualisiert. Sobald ich einen Fehler mache bleibt der Punktestand am entsprechenden Punkt stehen	Der Punktestand wird oben rechts angezeigt und laufend aktualisiert.	Testfall erfüllt
T8	Im ganzen Spiel war keine Werbung sichtbar	Während des Spielens und im Startmenü wird keine Werbung angezeigt.	Testfall erfüllt

#### 8.3.3 Testauswertung

Die Testprozedur verlief sehr gut wie erwartet. Alle acht Testfälle konnten erfolgreich abgeschlossen werden. Die Resultate der Testfälle deckten sich jeweils mit den erwarteten Ergebnissen. Somit müssen keine Nacharbeiten mehr erledigt und wir können unser Produkt mit gutem Gewissen dem Kunden übergeben.

## 8.4 Abnahme

Testdatum	17.05.16
Tester	Dominik Schütz
Gesamttestresultat	<input checked="" type="checkbox"/> Abgenommen <input type="checkbox"/> Abgenommen mit Nacharbeiten <input type="checkbox"/> Nicht abgenommen
Nacharbeiten	Keine Nacharbeiten erforderlich
Unterschrift Lieferant	
Unterschrift Kunde	

## 9 Einführungskonzept

### 9.1 Einführungsplan

Mit dem Einführungsplan soll beschrieben werden wie wir unser realisiertes Python Spiel auf dem produktiven Endgerät des Kunden installieren.

Da unser PyJump Spiel eine Neuentwicklung ist, sind keine Datenmigrationen notwendig. Aus Sicht des Endbenutzers müssen keine organisatorischen Abläufe angepasst werden, weil unser PyJump Spiel unabhängig von anderen Systemen lauffähig ist. Aus diesem Grund ist auch keine Pilotierung oder eine stufenweise Einführung nötig.

Die Installation von unserem PyJump Spiel geschieht äusserst einfach. Folgende Schritte müssen durchgeführt werden:

1. Software auf das Endgerät in das gewünschte Verzeichnis kopieren.
2. Jetzt muss sichergestellt werden ob die Python Image Library Pillow installiert ist. Dies kann man beispielsweise herausfinden, indem man das Spiel einfach mal startet. Falls Pillow fehlen würde erscheint eine entsprechende Fehlermeldung.
3. Es gibt mehrere Möglichkeiten das Spiel zu starten.
  - a. Über den Terminal mit `python main.py`
  - b. Über die Verknüpfung im Launchpad
4. Falls Pillow fehlen würde, muss zuerst der Python Packagemanager pip installiert werden. Dies geschieht über den Terminal mit: `sudo easy_install pip`
5. Jetzt kann Pillow über pip installiert werden mit: `sudo pip install Pillow`
6. Jetzt kann das Spiel wieder wie oben gestartet werden ohne dass eine Fehlermeldung erscheint.

#### Launchpad-Verknüpfung

Um das PyJump Spiel vom Launchpad oder vom Dock aus zu starten muss über den Automator (standardmässig vorinstalliert auf Mac) eine Applikation erstellt werden.

1. Automator starten und unter dem gewünschten Speicherort ein neues Dokument erstellen.
2. Danach wählt man Programm aus.
3. In der Suche „Shell-Skript ausführen“ eingeben und das Suchergebnis doppelklicken.
4. In der Textbox muss dann folgender Befehl eingetragen werden (Achtung Pfadanpassung): `python /path/to/installation/pyjump/main.py`
5. Je nach belieben kann das Icon der Verknüpfung mit einem Rechtsklick danach auf Informationen geändert werden, indem das gewünschte Icon per Drag n Drop über das alte gelegt wird.



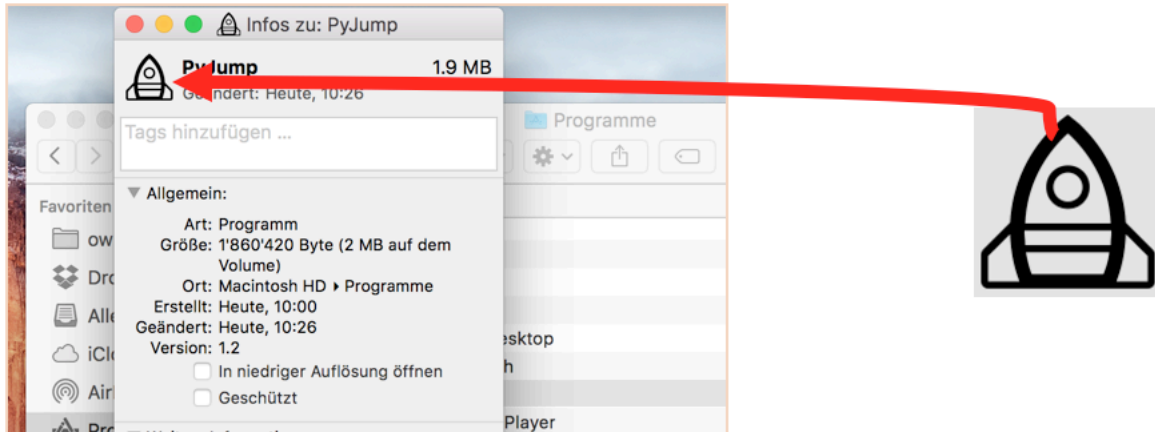


Abbildung 7 Icon ändern

## 9.2 Migrationsplan

Da bei diesem Projekt ein neues System erstellt wurde ist keine Migration notwendig.

## 10 Benutzerdokumentation

### 10.1 Benutzerhandbuch

#### 10.1.1 Systemübersicht

Diese Applikation stellt ein endlos Platformer dar. Das Ziel ist es mit der Spielfigur über mehrere Plattformen zu springen und so an höhe zu gewinnen. Der Spieler kann solange weiterspielen bis er einen Fehler macht und hinunterfällt. Je höher der Spieler springt ohne einen Fehler zu machen desto mehr Punkte erhält er.

Wie unter Abschnitt 2.1 Systemdesign ersichtlich ist wird das Spiel in mehrere Module aufgeteilt. Das hat den Vorteil, dass das Spiel einfacher erweitert werden kann. Die externen Schnittstellen werden im Abschnitt 2.2 Schnittstellendefinition beschrieben. Eine Schnittstelle wird dafür verwendet, dass der Spieler die Spielfigur steuern kann. Die andere Schnittstelle dient dazu, damit der Spieler im Menu Optionen auswählen kann.

Zur Sicherheit des Spiels muss man sich keine grossen Sorgen machen. Die Applikation wurde Malwarefrei programmiert und es werden keine sensiblen Daten gespeichert. Die einzigen Daten, die gespeichert werden sind die Scores des Spielers. Diese werden zusammen mit einem Nickname, der nicht zwingend echt sein muss gespeichert. Diese Daten werden allesamt lokal auf dem Gerät des Spielers gespeichert und bleiben auch dort, weil das Spiel über keine Verbindung zur Aussenwelt verfügt.

Als Anwenderrollen gibt es nur die des Spielers. Das Spiel kann nach der Bereitstellung ohne Technikwissen angewendet werden.

#### 10.1.2 Anwenderfunktionalität

##### Allgemein

Bei diesem Spiel handelt es sich um ein Endlos-Platformer. Das heisst das Spiel läuft solange weiter bis der Spieler einen Fehler macht oder das Spiel abbricht. Solange der Spieler ohne Fehler spielt wird der Punktezähler steigen. Wenn man kurz unterbrechen möchte, kann man mit der Taste „P“ das Spiel pausieren. Durch erneutes drücken der Taste „P“ wird das Spiel am gleichen Punkt weiterlaufen. Dieses Spiel hat keinen Lerninhalt. Es dient einzig und allein dem Unterhaltungszweck.

##### Installation

Da das Spiel in der Scriptsprache Python umgesetzt wurde ist eine Installation nicht nötig. Es reicht, wenn einfach alle benötigten Dateien an den gewünschten Ort kopiert werden. Gegebenenfalls muss jedoch ein Python-Modul nachinstalliert werden. Dafür muss zuerst mit Administratorrechten der Python Paketmanager pip installiert werden. Dies kann zum Beispiel über den Terminal mit dem Befehl „sudo easy\_install pip“ getan werden. Danach muss mit pip das Pillow-Modul installiert werden. Dies geschieht

wieder über den Terminal mit dem Befehl „sudo pip install Pillow“. Auch hierfür werden wieder Administratorrechte benötigt.

## Startmenü

Sobald man das Spiel startet wird einem als erstes das Startmenü angezeigt. Dort gibt es einen Abschnitt „Hilfe“, bei dem man sich Informationen über die Bedienung des Spiels holen kann. Zudem gibt es im Startmenü noch einen Abschnitt „Info“, wo man Informationen über die Version und die Entwickler des Spiels findet.

## Gameplay

Im Startmenü kann man das Spiel dann durch drücken der „Leertaste“ starten. Die Spielfigur fängt dann an zu springen sofern sie auf den Plattformen landet. Mit den Tasten „A“ und „D“ kann man die Spielfigur nach links oder rechts bewegen. Wenn man mit der Spielfigur am Rand das Fenster verlässt, so tritt die Spielfigur am anderen Rand wieder ins Fenster.

Während des Spielens können Monster auftreten, die die Spielfigur nicht berühren darf. Wenn die Spielfigur ein Monster berührt zählt das als Fehler und das Spiel wird beendet. Die Spielfigur kann sich aber natürlich auch gegen die Monster wehren. Wenn der Spieler die „Leertaste“ drückt, so kann die Spielfigur Kugeln gegen die Monster schießen. Wird ein Monster getroffen so verschwindet es aus dem Fenster und die Spielfigur kann vorerst ungehindert weiter springen.

## Problembehandlung

Falls beim Spielen Probleme auftauchen kann das Spiel unter Mac Computern mit der Tastenkombination „cmd“ + „Q“ oder über das Menü in der Symbolleiste beendet werden. Danach kann das Spiel wieder wie gewohnt gestartet werden. Falls sich das Spiel wie vorhin beschrieben über den üblichen Weg nicht beenden lässt kann mittels der Tastenkombination „cmd“ + „alt“ + „esc“ das „Sofort Beenden“-Menü aufgerufen werden. In diesem Menü kann man Python auswählen und dieses dann zum stoppen erzwingen. Danach kann das Spiel ebenfalls wie gewohnt wieder gestartet werden.

## Wiederherstellung

Das Spiel kann ohne grossen Aufwand wiederhergestellt werden, für den Fall das trotz neustarten des Spiels immer noch Probleme auftreten. Für die Wiederherstellung wird zuerst der ganze „pyjump“-Ordner gelöscht. Sobald alle alten Dateien vollständig gelöscht wurden kann die ursprüngliche Version wieder auf das Zielgerät kopiert werden.

Falls nach der Neuinstallation des Spiels immer noch Probleme bestehen kann es sein, dass Python ein Problem hat. Das Spiel wurde für Python Version 2.7 entwickelt. Es wird empfohlen diese Version zu installieren. Falls bereits Python 2.7 installiert ist kann eventuell eine Neuinstallation von Python das Problem lösen.

## 10.2 Supporthandbuch

### 10.2.1 Massnahmen bei Benutzerproblemen

Beim PyJump-Spiel gibt es eigentlich keine Benutzerprobleme. Es kann aber vorkommen, wie bei jeder Applikation, dass sich das Spiel aufhängt. Bei solchen Problemen ist das Vorgehen gleich wie bei anderen Applikationen. Das Spiel muss neugestartet werden. Dies kann mit der Tastenkombination „cmd“ + „Q“ oder „cmd“ + „alt“ + „esc“ getan werden.

Alle anderen Benutzerprobleme sind auf falsche Bedienung zurückzuführen. Um das zu verhindern müssen die Benutzer geschult werden. Im Abschnitt 3.2 Benutzerhandbuch ist die Bedienung des Spiels ausführlich beschrieben.

### 10.2.2 Massnahmen bei technischen Problemen

Falls es während des Spielens zu einem Problem kommen sollte, so kann der Benutzer das Spiel abbrechen und Neustarten. Falls das Problem weiterhin besteht sollte zuerst sichergestellt werden, dass das Spiel auch wirklich die Fehlerquelle ist. Falls das zutrifft kann eine Neuinstallation eventuell Abhilfe leisten.

Möglicherweise kann das Spiel nicht gestartet werden, weil das Pillow-Modul fehlt. Falls dies der Fall wäre würde eine entsprechende Fehlermeldung erscheinen. Um dies zu beheben muss zuerst mit

Administratorrechten der Python Packagemanager pip installiert werden. Dies kann zum Beispiel über den Terminal mit dem Befehl „sudo easy\_install pip“ getan werden. Danach muss mit pip das Pillow-Modul installiert werden. Dies geschieht wieder über den Terminal mit dem Befehl „sudo pip install Pillow“. Auch hierfür werden wieder Administratorrechte benötigt.

Falls nun immer noch Probleme auftauchen sollte überprüft werden ob Python richtig funktioniert. Falls dies nicht der Fall ist kann eventuell eine Neuinstallation von Python das Problem lösen. Zudem sollte beachtet werden, dass dieses Spiel für die Python Version 2.7 entwickelt wurde. Unter anderen Python Versionen kann sich das Spiel möglicherweise anders verhalten.

## 10.2.3 Anhang zum Suppothandbuch

### Glossar

Stichwort	Beschreibung
PIP	Paketverwaltungsprogramm für Python-Pakete
Pillow	Python Library für Bilder
sudo	Befehl mit Administratorrechten ausführen
easy_install	Pythonmodul um Pythonpakete zu installieren
Terminal	Kommandozeile
Python	Scriptsprache

## 10.3 Ausbildungsplan

Für unser PyJump Spiel ist keine besondere Ausbildung nötig. Das Spiel und die Installation sind sehr simpel gehalten. Sobald man das Spiel startet erscheint im Startmenu die Spielanleitung. Als Installationsanleitung sollten die Erläuterungen in diesem Dokument reichen.

## 11 Zielerreichung

### Ziele des Initialisierungsantrags

Nr	Ziel	Bewertung
Z1	Durch einen sauberen Initialisierungsantrag wird die Initialisierungsphase freigegeben.	✓
Z2	Die Stärken und Schwächen der IST-Situation werden ausführlich festgehalten.	✓
Z3	Die Ziele sind strukturiert und entschärfen die Schwächen der IST-Situation.	✓
Z4	Die Projektziele beziehen sich auf die SOLL-Situation.	✓
Z5	Die Anforderungen sind von den Zielen abgeleitet.	✓
Z6	Die Lösungsvarianten sind detailliert beschrieben und decken die Anforderungen ab.	✓
Z7	Die Lösungsvarianten werden gegenübergestellt, sodass der Lösungsentscheid nachvollziehbar ist.	✓

## Ziele der Studie

Nr	Ziel	Bewertung
Z1	Das Spiel ist ohne Malware.	✓
Z2	Das Spiel ist offline verfügbar.	✓
Z3	Das Spiel ist ohne zusätzlich installierte Software lauffähig.	x
Z4	Das Spiel läuft endlos weiter bis der Spieler einen Fehler macht.	✓
Z5	Dem Spieler wird der Punktestand laufend angezeigt.	✓
Z6	Das Spiel ist werbefrei.	✓

## Erläuterung zu Ziel Z3 der Studie

Beim Testen des PyJump-Spiels auf einem anderen MacBook fiel uns auf, dass sich das Spiel nicht starten lässt, weil ein Python-Modul fehlte. Auf unserem Entwicklungsgerät war dieses Modul bereits installiert, weshalb wir diese Abhängigkeit zu spät entdeckten um sie noch zu umgehen. Das fehlende Python-Modul heisst Pillow und ist eine Image-Library. Pillow kann mit wenig Aufwand über die Kommandozeile wie folgt installiert werden:

- `sudo easy_install pip`
- `sudo pip install Pillow`

## 12 Erfahrungen

### 12.1 Persönliche Erfahrungen von Dominik Schütz

Vorweg würde ich sagen, dass uns dieses Projekt im grossen und Ganzen gut gelungen ist. Am Anfang des Projekts war es für mich noch etwas ungewohnt in der Rolle des Projektleiters. Ich musste mich um sehr viele Angelegenheiten kümmern wie das Zeitmanagement, das Einhalten der Termine, das versenden von Statusupdates und natürlich um einen grossen Teil der Dokumentation. Ich habe es unterschätzt all das im Überblick zu behalten, was man auch merkte. Die Statusupdates, welche ich eigentlich jeden Sonntag verschickte blieben circa zweimal aus. Mit dem Start der Initialisierungsphase habe ich mich dann aber in die Rolle des Projektleiters eingelebt, sodass die weiteren Statusupdates alle pünktlich versendet wurden.

Ein grosser Misserfolg war auch der Initialisierungsantrag. Aufgrund eines Missverständnis haben wir diesen eine Woche zu spät abgegeben, obwohl er inhaltlich fertig war. Dies hat uns einen grossen Punkteabzug gebracht. Um solche Missverständnisse für spätere Abgabetermine von Dokumenten zu verhindern, habe ich mir dann alle Termine im Kalender eingetragen. Zudem habe ich periodisch geprüft, ob sich ein Abgabetermin möglicherweise verändert hat. Dieses Vorgehen hat sich ausgezahlt. Alle anderen Abgabetermine konnten wir einhalten, sodass wir den Punkteabzug vom Initialisierungsantrag kompensieren konnten.

#### 12.1.1 Was habe ich gelernt?

Dieses Projekt war für mich auf jeden Fall sehr lehrreich. Einerseits im Bezug aufs Projektmanagement und andererseits im Bezug auf die Python-Entwicklung.

Ich habe vorher noch nie in einem solchen Projekt mitgearbeitet, deshalb war alles für mich neu und ich konnte bei der Bearbeitung von jedem Dokument etwas lernen. Ich lernte unter anderem, wie man eine Situationsanalyse durchführt sprich wie man die Ausgangslage beschreibt und dann einen Soll/Ist Vergleich macht. Ausserdem lernte ich wie man verschiedene Umsetzungskonzepte erstellt, diese bewertet und die beste Lösung auswählt. Zudem lernte ich, wie man aus dem Umsetzungskonzept dann eine Systemarchitektur entwirft und für diese ein Testkonzept und Testfälle entwickelt.

Da wir aus der Systemarchitektur ja dann unser Spiel entwickelten lernte ich wie man objektorientiert Python programmiert. Zudem lernte ich einige Python-Module kennen, die wir zum Beispiel für das GUI verwendeten. Als wir unser PyJump-Spiel fertig entwickelt hatten lernte ich, wie man einen Einführungsplan erarbeitet. Daraus ergab sich dann auch die Notwendigkeit einer Benutzerdokumentation. Somit lernte ich wieder etwas Neues.

## 12.1.2 Was verlief gut?

Bis auf ein paar Ausnahmen verlief das gesamte Projekt gut. Bei der Studie, dem Konzept- und Realisierungsbericht konnten wir jeweils viele Punkte holen. Die Realisierung verlief ebenfalls nach Plan und ohne grössere Zwischenfälle. Besonders freute es mich, dass alle Testfälle gleich beim ersten Versuch erfolgreich waren, sodass wir keine Nacharbeiten erledigen mussten.

Gut verlief auch die Einführungsphase. Unser PyJump-Spiel konnten wir ohne Probleme auf dem Kunden-MacBook installieren und es lief wie erwartet.

## 12.1.3 Was verlief weniger gut?

Wie im obigen Abschnitt bereits erwähnt gab es nicht viele Dinge, die schlecht liefen. Trotzdem gab es wie in jedem Projekt ein paar Misserfolge. Direkt am Anfang des Projekts erhielten wir einen grossen Punkteabzug, weil wir aufgrund eines Missverständnisses den Initialisierungsantrag eine Woche zu spät abgaben, obwohl wir diesen eigentlich schon fertig hatten.

Zudem konnten wir eine Anforderung nur zum Teil erfüllen, weil wir in unserem PyJump-Spiel ein Python-Modul verwendeten, welches unter Umständen nicht standardmässig bei jeder Python-Installation dabei ist. Dies bemerkten wir erst nach der Realisierung, weil auf unserem Entwicklungsgerät diese Python-Modul bereits installiert war.

Nach der Installation des fehlenden Python-Moduls war unser PyJump-Spiel auch auf dem Kundengerät lauffähig. Das fehlende Modul namens Pillow ist eine Image Library und wurde zwingend benötigt, damit unser PyJump-Spiel überhaupt gestartet werden konnte.

## 12.1.4 Was würde ich in Zukunft anders machen?

Beim nächsten Projekt würde ich auf jeden Fall die Termine besser planen, sodass es keine Missverständnisse gibt. Wenn ich das bei diesem Projekt auch schon so gemacht hätte dann hätten wir für den Initialisierungsauftrag keinen Punkteabzug erhalten.

Ebenfalls werde ich mir ab jetzt immer als erstes einmal die Bewertungskriterien anschauen, damit ich weiss wo ich besonders sorgfältig vorgehen muss.

Ausserdem ist es für ein nächstes Projekt sicherlich von Vorteil, wenn bereits während der Realisierungsphase überprüft wird welche Abhängigkeiten bestehen. Damit so etwas wie ein fehlendes Python-Modul nicht wieder vorkommt.

## 12.2 Persönliche Erfahrungen von Raphael Schwob

Ich denke im Grossen und Ganzen können wir zufrieden sein mit unserem Projekt. Auch wenn der Fokus mehr auf der Dokumentation lag als auf dem Code selbst, konnte ich hier und da ein paar neue Dinge dazulernen im Bezug auf die Arbeit mit Python. Am Ende hatte das Game zwar immer noch Potential für Verbesserungen, da aber der Fokus eben nicht auf einem Perfekten Code lag, haben wir diesen am Schluss auch nicht mehr gross angepasst.

Durch dieses Projekt und die Wahl der Entwicklungsumgebung, konnte ich auch ein wenig was lernen, das ich dann im Betrieb auch wirklich brauchen kann. In meiner Firma arbeiten wir mit dem Python Framework/CMS Plone. Durch die Wahl von Python als Programmiersprache hatte ich natürlich bereits im Voraus einige Vorteile, da ich mich schon ein wenig damit auskannte. Trotzdem sind wir hier und da wieder auf Probleme gestossen, die auch ich nicht gleich zu lösen wusste. Dadurch konnte ich doch einiges Neues dazulernen.

Doch hauptsächlich haben wir beide gute Erfahrungen mit dem Dokumentieren eines solchen Projektes machen können. Ich denke als Vorbereitung auf die IPA im nächsten Jahr war dieses Projekt eine super

Sache. Wir wissen nun beide wie man eine Doku richtig verfassen würde und könne das Gelernte bei der IPA sicher sehr gut wiederverwenden und die Doku die wir nun für dieses Projekt verfasst haben wird sicher als ein gutes Beispiel helfen.

## 12.2.1 Was habe ich gelernt?

Wie bereits erwähnt, habe ich sicher sehr viel über Projektdokumentationen lernen können, aber hier und da auch wieder ein paar Dinge über die Python Entwicklung. Der Fokus des ganzen Projektes lag aber ganz klar auf der Dokumentation, und dort konnte ich durch das Schreiben der Doku natürlich sehr viel Neues dazulernen. Ich denke mit dem Gelernten bin ich nun recht gut auf die IPA vorbereitet. Wenn ich an dem Projekt in einem halben Jahr das angeeignete Wissen richtig anwenden kann, sollte ich sicher einen guten Lehrabschluss machen können. Diese Doku wird mir sicher auch hier und da als Vorlage/Beispiel dienen, da sie wirklich sehr umfangreich ist.

Vom Anfang bis zum Schluss haben wir wirklich jeden einzelnen Schritt in der Doku festgehalten. Durch die jeweiligen Vorlagen und Beispiele aus den Moduldokumenten wussten wir immer ziemlich gut was genau wir machen mussten.

## 12.2.2 Was verlief gut?

Aus meiner Sicht verlief das Schreiben des Codes recht gut. Ich vermute, das war hauptsächlich, weil wir bereits im Voraus alles perfekt dokumentiert hatten und genau wussten, was das Game am Ende alles können sollte. Durch die Auflistung aller Anforderungen mussten wir nicht mehr gross überlegen, was das Game jetzt alles braucht. Die Strukturierung war auch schon ziemlich detailliert vorgegebenen und ich musste eigentlich nur noch die Doku in Code umschreiben und nicht noch gross überlegen wie jetzt eine bestimmte Funktion implementiert werden muss.

Was auch gut geklappt hat war die Organisation im Team. Das ist natürlich hauptsächlich dem Projektleiter zu verdanken. Er hat einen grossen Teil der Doku geschrieben und hat mir immer gut kommuniziert wann ich etwas fertig haben musste oder wie das weitere Projekt aussieht.

## 12.2.3 Was verlief weniger gut?

Gerade am Anfang habe wir einen Abgabetermin verpasst, da wir ein kleines Missverständnis mit der Lehrperson hatten. Wir hatten das Dokument zwar komplett fertig geschrieben, dachten jedoch, dass wir dieses erst eine Woche später abgeben müssen, wodurch wir den Abgabetermin leider verpasst hatten. Danach hatten wir aber kaum noch Probleme die Termine einzuhalten. Ich glaube unser Projektleiter hat sich nach diesem Problem direkt alle Termine im Kalender eingetragen, dass sowas auch sicher nicht mehr vorkommt.

## 12.2.4 Was würde ich in Zukunft anders machen?

Grundsätzlich gibt es nicht viel, was ich beim nächsten Mal anders machen würde. Was man aber sicher besser machen könnte ich die Sache mit der Terminplanung. Beim nächsten Projekt sollte man sicher von Anfang an besser abklären wann welches Dokument abgegeben werden sollte damit nicht wieder irgendwelche Abgabetermine verpasst werden und es dadurch zu einer schlechteren Gesamtwertung kommt.

## 13 Persönliche Vorsätze für IPA

### 13.1 Vorsätze von Dominik Schütz

- Für die IPA will ich ein Thema finden, welches mich interessiert, mir Freude bereitet und vom Umfang her für eine IPA geeignet ist.
- Ich will eine Dokumentation schreiben, welche alle Aspekte des Projekts abdeckt und den Anforderungen der IPA entspricht.
- Ich will eine Präsentation durchführen, welche die Experten von meiner Arbeit überzeugt.

## **13.2 Vorsätze von Raphael Schwob**

- Ich werde versuchen, so weit es möglich ist, mir ein Projekt auszusuchen, welches mich interessiert. Am besten in einem Themenbereich der mir nicht komplett fremd ist.
- Am Ende des Projekts möchte ich eine Präsentation machen, welche alle Aspekte des Projekts aufzeigt.
- Ich will eine Dokumentation schreiben, die das Projekt so umfangreich wie nur möglich beschreibt.

## 14 Glossar

Stichwort	Beschreibung
PyJump	Name des Spiels, welches in diesem Projekt realisiert wurde
Python	Skriptsprache
PIP	Paketverwaltungsprogramm
easy_install	Paketverwaltungsprogramm
Pillow	Python Image Library
Terminal	Kommandozeile
sudo	Befehl um ein Programm mit Administratorrechten zu starten
GUI	Grafische Benutzerschnittstelle
Malware	Schadsoftware
Automator	Programm für den Mac
Icon	Kleines Symbolbild
Drag n Drop	Technik um mit der Maus etwas zu verschieben
Shell-Skript	Skript, welches von einer Shell ausgeführt werden kann
Launchpad	Applikationsübersicht auf dem Mac
Java	Programmiersprache
Red Hat	Informatik Firma
Red Hat Standard Subscription	Abonnement für ein Red Hat Server Betriebssystem
Javascript	Skriptsprache im Browser
PHP	Server Skriptsprache
Unix	Betriebssystem
Doodle Jump	Smartphone-Spiel