Konzeptbericht

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

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
1.0	08.03.16	Erstellung dieses Dokuments	Dominik Schütz
1.2	13.03.16	Weiterführung des Dokuments	Raphael Schwob
2.0	15.03.16	Letzte Überarbeitungen	Raphael Schwob, Dominik Schütz

Definitionen und Abkürzungen

Begriff / Abkürzung	Bedeutung
Ransomware	Malware, die die Festplatte verschlüsselt.
Tkinter	Python Package für GUI
GUI	Graphical User Interface
Usecase	Bündelt mögliche Szenarien, die Eintreten könnten.
Spawn-Prozess	Erstellung von Spielelementen.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Zusammenfassung

Die in der Initialisierungsphase gewählte Lösungsvariante wir in diesem Dokument konkretisiert. Die Lösungsvariante soll so detailliert beschrieben werden, dass das Produkt verlässlich geplant und realisiert werden kann.

In diesem Dokument werden die Anforderungen an Funktionalität, Informationssicherheit und Datenschutz festgehalten. Zudem wird die Systemarchitektur erläutert. Diese beinhaltet unter anderem eine Übersicht über die Module, welche für diese Lösungsvariante benötigt werden und über die benötigten Schnittstellen, die zwischen den einzelnen Modulen (intern) und zwischen System und Aussenwelt (extern) zum Einsatz kommen. Des weiteren wir in diesem Dokument ein Testkonzept erarbeitet, welches sicherstellt, dass unser Produkt nach der Realisierungsphase ausgiebig getestet wird.

Am Ende dieses Dokuments wird auf die Weiterführung der Projektplanung eingegangen. Es wird der aktuelle Stand des Projekts analysiert und die Risikoanalyse wird aktualisiert. Zum Schluss wird die aktualisierte und detaillierte Planung der Realisierungsphase aufgezeigt.

2 Systemanforderungen

2.1 Anforderungen an die Funktionalität

Teilanforderung	Beschreibung
A1.1	Das Spiel darf keine Malware enthalten. Dazu zählen insbesondere Würmer, Viren, Trojaner, Spyware und Ransomware.
A1.2	Das Spiel muss aus einer vertrauenswürdigen Quelle stammen.
A2	Das Spiel muss offline gespielt werden können. Damit wird gemeint, dass das Spiel eine Verbindung zum Internet oder zu einem anderen Netzwerk nicht benötigt.
A3	Das Spiel muss ohne zusätzlich installierte Software lauffähig sein. Sprich es darf keine Laufzeitumgebung zusätzlich installiert werden.
A4.1	Das Spiel darf an keine Höchstpunktzahl gebunden sein.
A4.2	Das Spiel läuft endlos weiter bis der Spieler einen Fehler macht.
A5.1	Der Punktestand muss laufend aktualisiert werden.
A5.2	Der Punktestand muss dem Spieler laufend angezeigt werden.
A6	Das Spiel muss komplett werbefrei sein.

2.2 Aufgaben decken Anforderungen ab

Aufgabe	Beschreibung	Anforderung
a1	Das Spiel muss sorgfältig programmiert werden. Zudem muss darauf geachtet, werden, dass niemand Malware in den Code einschleusen kann.	A1.1 A1.2
a2	Das Spiel muss unabhängig von einer Internetverbindung lauffähig sein.	A2
a3	Das Spiel muss nur auf das Gerät des Kunden kopiert werden. Danach kann es ohne zusätzliche Konfiguration gestartet werden	A3
a4	Das Spiel läuft endlos weiter bis der Spieler einen Fehler macht.	A4.1 A4.2
a5	Solange der Spieler keinen Fehler gemacht hat und weiter spielt muss der Punktestand laufen neu berechnet und aktualisiert werden	A5.1
a6	Es muss ein Feld existieren, wo dem Spieler der aktuelle Punktestand angezeigt wird.	A5.2
a7	Das Spiel darf keine Werbung enthalten.	A6

2.3 Anforderungen an die Informationssicherheit und den Datenschutz

In diesem Abschnitt werden organisatorische und technische Massnahmen zur Sicherstellung von Verfügbarkeit und Authentizität der Daten als Anforderungen formuliert. Zudem werden die Anforderungen an den Schutz sensibler Daten vor unbefugtem Zugriff und vor missbräuchlicher Verwendung definiert.

Informationssicherheit

Anforderung	Beschreibung	Massnahme
A1	Der Quellcode für das PyJump-Spiel muss gesichert werden.	Der Quellcode wird durch den Head of Development auf einem externen Medium gesichert. Dabei werden die verschiedenen Versionen unterschieden.
A2	Projektdokumente müssen gesichert werden.	Die jeweils aktuellen Versionen der Projektdokumente werden durch den Projektleiter auf einem externen Medium gesichert.
A3	Bei einem technischen Defekt des Kundengeräts muss das PyJump-Spiel auf dem Ersatzgerät des Kunden installiert werden können.	Das PyJump-Spiel wird auf einem externen Medium gesichert und kann von dort aus auf das Ersatzgerät des Kunden installiert werden.
A4	Fahrlässiges Handeln der Mitarbeiter darf nicht zu Datenverlust führen.	Unsere Mitarbeiter arbeiten sehr sorgfältig. Zudem werden regelmässig Backups auf externen Medien erstellt.
A5	Dokumentationen dürfen nicht für jedermann ersichtlich sein.	Die Dokumentationen werden auf einem Medium mit Zugangskontrolle gesichert.

Datenschutz

Anforderung	Beschreibung	Massnahme
A1	Persönliche Kundendaten müssen geschützt werden.	Unsere Mitarbeiter wurden im Umgang mit sensiblen Daten geschult. Sensible Daten werden jeweils auf einem Medium mit Zugangskontrolle gesichert.
A2	Sensible Daten müssen vor Missbrauch geschützt werden.	Sensible Daten werden auf einem Medium mit Zugangskontrolle gesichert. Zudem haben unsere Mitarbeiter eine Vertraulichkeitserklärung unterschrieben.

3 Systemarchitektur

Die einzelnen Module werden in Python Klassen unterteilt. Dabei wird jeweils darauf geachtet, dass Objekte die im Spiel mehrmals in verschiedenen Ausführungen auftauchen jeweils durch eine Hauptklasse gesteuert werden.

Als Grundlage für dieses Prinzip wird die main Klasse verwendet. Sie steuert durch mehrere Funktionen das Verhalten des Spieles und delegiert einzelne Aufgaben an die jeweilige Subklasse. Als Beispiel für dieses Prinzip kann man das Zeichnen des Spielfeldes nehmen. Die Main Klasse ist hierbei das Spielfeld selbst, d.h. die Klasse erbt direkt von der Canvas Klasse vom tkInter. Deshalb verwaltet die Hauptklasse auch direkt alles was auf dem Spielfeld passiert. Allerdings die Bewegungen der einzelnen Elemente auf dem Spielfeld, wie z.B. der Plattformen oder der Monster werden durch die jeweiligen Klassen gesteuert.

Die Player Klasse ist bei diesem Prinzip ganz alleine für die Eingaben des Spielers verantwortlich, die Hauptkasse ist also nicht auf den Userinput angewiesen und muss nur die Objekte auf dem Spielfeld jeweils dorthin bewegen, wo es die Subklassen sagen.

Als einzige Schnittstelle zwischen den einzelnen Modulen dient die Main Klasse. Alle Module sind in ihr vorhanden, dadurch kann die Hauptklasse Aufgaben zwischen den einzelnen Modulen koordinieren und falls nötig können auch die einzelnen Module über die Schnittstelle miteinander kommunizieren.

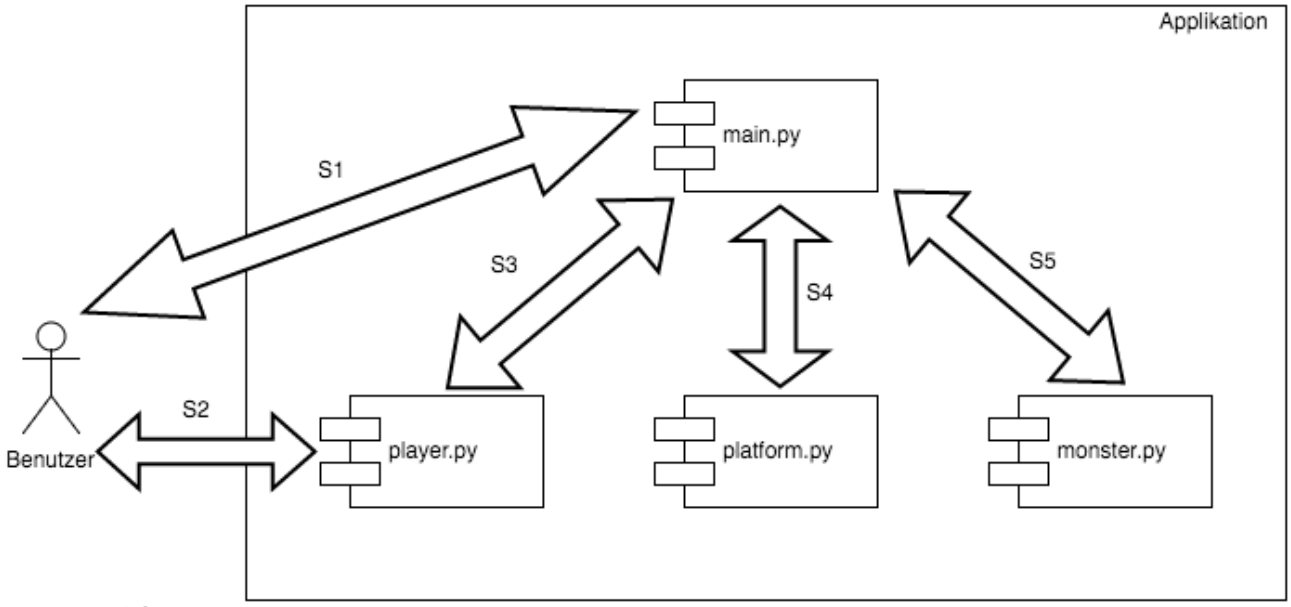

Abbildung 1 Systemarchitektur

3.1 Gliederung der Lösung in Module

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 Usereingaben 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.

3.2 Schnittstellen

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 wie z.B. der Schwierigkeitsgrad angepasst werden kann. 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.

4 Testkonzept

4.1 Systemtest

Die Tests werden per Usecases definiert. Die Usecases orientieren sich jeweils an den einzelnen Features die das Game enthalten soll. Aufgrund der Struktur die im Python verwendet werden kann, kann das ganze Game laufend um einzelne Features erweitert werden. Sobald diese dann einmal implementiert wurden, wird jeweils ein Usecase dafür erfasst. Bei jeder neu hinzugefügten Erweiterung müssen dann die Usecases der vorher implementierten Features wiederum getestet werden. Dabei sollten die einzelnen Features möglichst unabhängig voneinander sein, damit das modulare Erweitern, wie es von Python vorgesehen ist, auch am Ende des Projekts immer noch funktioniert.
Falls also ein Usecase nach der Implementation eines neuen Features nicht mehr erfolgreich ist, so sollte die Dependency die entstanden ist gefunden und entfernt werden. Durch diesen Prozess kann verhindert werden, dass bei zukünftigen Problemen eine Kettenreaktion im Code auftritt wird.
Ein Beispiel dafür wäre, wenn ein neuer Typ von Monster hinzugefügt wird, dadurch aber plötzlich die Plattformen durch eine Dependency im Spawn Prozess nicht mehr richtig angezeigt werden. Dies sollte auf keinen Fall möglich sein, weshalb es sehr wichtig ist, dass die einzelnen Module so unabhängig wie möglich voneinander sind.

4.2 Abnahmetest

Der Abnahmetest sollte jeweils nach einem erfolgreichen Systemtest gemacht werden. Sobald ein neues Feature erfolgreich implementiert wurde, sollte getestet werden, ob das Game auch auf einem anderen System ohne die Installation von zusätzlichen Paketen funktioniert. Sollte die Installation eines zusätzlichen Paketes notwendig sein, muss geprüft werden, ob das Feature auch ohne dieses Paket umgesetzt werden kann. Sollte das nicht möglich sein, so muss geschaut werden, ob das Feature sehr wichtig ist. Falls das Feature wichtig genug ist, müsste ein Konzept zum automatischen Download von zusätzlichen Python Paketen umgesetzt werden. Dies ist grundsätzlich möglich, allerdings ist es unser Ziel wenn möglich ohne zusätzliche Pakete auszukommen.

4.3 Testfälle

Testfall	Beschreibung	Erwartetes Resultat	Anforderung
T1	Das Netzwerkkabel wird gezogen und WLAN wird ausgeschaltet. Danach wird das Spiel gestartet.	Das Spiel muss starten und weiterlaufen, bis der Spieler einen Fehler macht.	A2
T2	Das Spiel wird auf das Gerät des Kunden kopiert und wird gestartet.	Das Spiel muss starten und weiterlaufen, bis der Spieler einen Fehler macht.	A3
T3	Das Spiel wird gestartet und der Spieler fängt an zu spielen.	Das Spiel muss weiterlaufen, bis der Spieler einen Fehler macht, sprich eine Plattform verfehlt.	A4.1 A4.2
T4	Das Spiel wird gestartet und der Spieler fängt an zu spielen.	Während dem Spiel aktualisiert sich der Punktestand fortlaufen und wird dem Spieler angezeigt.	A5.1 A.2
T5	Das Spiel wird gestartet und der Spieler fängt an zu spielen.	Im Menu sowie während dem Spiel wird keine Werbung angezeigt.	A6

5 Weiterführung der Projektplanung

5.1 Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept

Der Soll/Ist-Vergleich zwischen Zeitplanung und effektivem Verlauf der Konzeptphase sieht gut aus. Die Planung deckt sich mit dem Verlauf der Konzeptphase. Grund dafür ist auch, dass keine Schwierigkeiten auftauchten. Es traten keine befürchteten Risiken ein.

5.2 Aktualisierung der Risikosituation

An der Risikosituation hat sich nichts verändert.

Risiko	Beschreibung	Massnahme
R1	Verlust der Projektdokumente durch technische Fehler.	Es werden regelmässig Backups erstellt.
R2	Verlust der Projektdokumente durch menschliches Versagen.	Es werden regelmässig Backups erstellt.
R3	Missbrauch der Daten durch unberechtigten Zugriff.	Die Dokumente werden durch Zugangskontrollen geschützt.

5.3 Planung der nächsten Phase

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

Abbildung 2 Planung Realisierungsphase