Jump-And-Run

Software Development Plan

1. Vorwort 3

1.1 Ziel und Umfang 3

1.2 Referenzen und Delegation 3

2. Projektübersicht 4

2.1 Projektziele, Annahmen 4

3. Projektorganisation 7

3.1 Struktur des Projektteams 7

3.2 Rollen und Verantwortlichkeiten 7

4. Managementprozess 10

4.1 Projektplan 10

4.1.1 Phasen Plan 10

4.1.2 Iterationszyklen 11

4.1.3 Releases 14

1. Vorwort

1.1 Ziele und Umfang

Dieses Dokument beschreibt den Rahmen und die Struktur, welche bei der Entwicklung der Software „Jump-and-Run“ Spiel erzeugt und eingehalten werden müssen. Es umfasst dabei den Arbeitsrahmen des gesamten Entwicklungsprozesses, sowie die Strukturierung, des Teams und Arbeitsablaufs. Das Dokument kann nur über Rücksprache mit dem Team geändert werden.

1.2 Referenzen und Delegation

* Glossar - Definitionen, Akronyme, und Abkürzungen

2. Projektübersicht

2.1 Projektziele, Annahmen und Einschränkungen

Das Ziel des Projektes „Jump-and-Run“ ist es ein Flappy-Bird ähnliches Spiel zu entwickeln welches möglichst Plattformunabhängig programmiert werden soll um die Möglichkeit eines Ports auf andere Systeme zu gewährleisten. Das Spiel soll dabei nicht getreu seinem Vorbild gebaut werden sondern ihm ähneln, sich dennoch so weit unterscheiden, dass es sich von ihm abhebt.

3. Projektorganisation

3.1 Struktur des Projektteams

Das Projektteam besteht aus 6 Mitgliedern in dem jedes Mitglied mehrere Rollen übernehmen muss. Bei internen Abstimmungen ist jedes Mitglied gleich gestellt.

3.2 Rollen und Verantwortlichkeiten

|  |  |
| --- | --- |
| Person | Rolle |
| Angirillo, Fabio | GUI-Designer Tester  Programmierer |
| Keller, Illja | Art Director  Tester  Anforderungsspezialist/Analyst |
| Lengler, David | Tester  Dokumentenverantwortlicher  Programmierer |
| O'Kagu, Kenny | Music Director  Tester  Dokumentenverantwortlicher |
| Stein, Patrick | Programmierer  Tester |
| Warmuth, Niklas | Projektleiter  Programmierer  Softwarearchitekt |

|  |  |
| --- | --- |
| Rolle | Beschreibung |
| Projektleiter | Leitung des Projektes:  Koordination, Planung, Steuerung, Kontrolle  Aufgaben & Kompetenzen:  Verantwortlichkeit für Gesamtprojekt  Weisungsbefugnis  Überblick über Kosten-, Ressourcen-, Zeiteinhaltung  Delegation von Aktivitäten |
| GUI-Designer | Ermittlung der grafischen Eigenschaften für die Oberfläche  Aufgaben & Kompetenzen:  Anordnung der grafischen Komponenten  Bedienung des Programms  Software-Ergonomie  Gedanken über Animationen und Grafiken  Erstellung eines grafischen Modells |
| Software-Architekt | Umsetzung der Anforderungen in ein Modell zur Implementierung  Überwachung und Beratung beim Umsetzungsprozess von der Vision bis zur Implementierung  Aufgaben & Kompetenzen  Beziehung zwischen Modulen erkennen  Umsetzung textuelle Anforderungen in Modelle  Umgang mit UML-Werkzeugen  Überwachung des Gestaltungsprozess  Koordination der technischen Aktivitäten  Überwachung der Restriktionen bei der Umsetzung  Erfahrung mit UML, Programmierung und Design von Softwareprodukten  Enge Zusammenarbeit mit Softwaredesigner |
| Qualitätsmanager | Beobachtung der Einhaltung von Qualitätsansprüchen an die Software  Aufgaben & Kompetenzen  Entwicklung von Testfällen und Testfunktionen  Oberflächentests  Modultests  Integrationstests  Abnahmetests |
| Tester | Qualitätssicherung der Softwarefunktionen und Module  Aufgaben & Kompetenzen  Erstellung Testfallpläne und Abarbeitung  Umgang mit J-Unit  Überprüfung auf semantische Fehler |
| Anforderungsspezialist/  Analyst | Entwicklungen von Anforderungen für Projekt zur zufriedenstellenden Erreichung des Ziels  Aufgaben & Kompetenzen  Analyse des erwünschten Projektergebnisses  Erstellung von Use Cases  Vergleich der Anforderungen mit der Umsetzung unter Berücksichtigung von Restriktionen  Erstellung von Change-Requests |
| Programmierer | Implementierung der Funktionalitäten  Aufgaben & Kompetenzen  Erstellung Quellcode aus UML-Modellen  Realisierungstechniken: XP, Einzelarbeit  Objektorientierte Programmierung  Kenntnisse in JAVA   * Algorithmen & Datenstrukturen * AWT, SWING * JDBC * … |
| Dokumentationsverantwortlicher | Managen und Realisierung der Dokumentations-erstellung bzw. Bereitstellung benötigter Vorlagen  Aufgaben & Kompetenzen  Erstellung von Formatvorlagen für Projekt  Festhaltung der wichtigsten Entscheidungen  Erstellung von Formblättern & Richtlinien  Verwaltung der Dokumente |
| Art Director | Bereitstellung sowie Design benötigter Graphischer Artefakte. |
| Music Director | Bereitstellung sowie Design benötigter Musikalischer Artefakte. |

4. Managementprozess

4.1 Projektplan

4.1.1 Phasenplan

|  |  |  |
| --- | --- | --- |
| Phase | No. of Iterations | Duration |
| Inception Phase | 1 | 3 Weeks |
| Elaboration Phase | 1 | 3 Weeks |
| Construction Phase | 3+1 | 8 Weeks |
| Transition Phase | 1 | 3 Weeks |

4.1.2 Iterationszyklen

Im Folgenden werden die einzelnen Iterationen in den Phasen näher beschrieben.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Iteration | Beschreibung | Meilensteine | Risiken \* |
| Inception | Inception | Festlegung des Projektziels und des Projektumfeldes in einer Vision.  Festlegung des Projekt-managementprozesses.  Aufstellung und Benennung der Anforderungen an die Software mittels Use-Cases.  Planung der Projektumsetzung; Verteilung und Festlegung von Aufgaben und Rollen.  Risikoanalyse mit Präventivmaßnahmen | Anforderungs-Dokument  Prototypen  End-Review:  Inception |  |
| Elaboration | Elaboration | Konzeptionelles Design für Oberfläche und Bedienung.  Erstellung des Softwaremodells  Erzeugung eines Architektur Prototypen | Review: GUI-Akzeptanz  Aufbau- & Ablauf-modellierung  Architektur Prototypen |  |
| Construction | C1 | Design und Implementierung der Beta Use Cases  Vorbereitung und Planung des Android Ports  Qualitätssicherung durch Modultests, Systemtests | Review:  Quellcode  Review:  Changerequests  beta-Protoyp |  |
|  | C2 | Design und Implementierung der Basic Use Cases  Integrationsanpassungen  Qualitätssicherung durch Modultests, Integrationstests, Systemtests | Review:  Quellcode  Review:  Changerequests  Rel.1.0 |  |
|  | C3 | Design und Implementierung der Advanced Use Cases  Qualitätssicherung  durch Testverfahren | Review:  Quellcode  Review:  Changerequests  Rel.2.0 |  |
|  | C4  Optional | Design und Implementierung der Expert Use Cases  -und/oder-  Design und Portierung auf Android  Qualitätssicherung durch benötigte Testverfahren | Review:  Quellcode  Review:  Changerequests  Rel.3.0 |  |
| Transition | Übergabe Softwareprodukt | Anfertigung Produkthandbuch & Softwarepräsentation  Abschließender Akzeptanztest, ggf. Fehlerbehebung  Eventuell Upload zum Google Play Store | Dokumentation  Review: Projekt-abschluss |  |

4.1.3 Releases

Das Release 1.0 welches in der 2. Iteration entsteht enthält alle Standardmäßig in einem Flappy-Bird Spiel vorhandenen Menüs sowie Fähigkeiten, außer irgendeine Form von Netzwerkanbindung.

In Release 2.0 werden diese dann hinzugefügt, sodass nun Gobale Highscores hinzugefügt werden und weiterhin es Möglich ist den eigenen Highscore zu teilen. Zudem werden dem Spiel neue Elemente hinzugefügt – wie z. B. Das aufheben und verwenden von Items.

In Release 3.0 werden weitergehende Features hinzugefügt, wie z.B. die Fähigkeit einen Account zu besitzen und die Items und Highscores darauf speichern zu können.