SOFTWARE ENGINEERING 2

PROJEKTMAPPE

„BLOKUS“

Inhaltsverzeichnis

**Geben Sie die Kapitelüberschrift ein (Stufe 1)1**

Geben Sie die Kapitelüberschrift ein (Stufe 2)2

Geben Sie die Kapitelüberschrift ein (Stufe 3)3

**Geben Sie die Kapitelüberschrift ein (Stufe 1)4**

Geben Sie die Kapitelüberschrift ein (Stufe 2)5

Geben Sie die Kapitelüberschrift ein (Stufe 3)6

# PROJEKTTEAM

## Mitglieder und deren Tätigkeiten

LAURA KOREN (1247748)

*Informationsmanagement, Semester 6*

*Dokumentation*

TOBIAS HOHENWARTER (1460443)

*Angewandte Informatik, Semester 4*

*Projektmanager*

DANIEL MANDL (1260078)

*Angewandte Informatik, Semester 6*

*Implementierung*

MARKUS MENNEL (1460236)

*Angewandte Informatik, Semester 4*

*Design*

MANUEL SEYWALD (1361416)

*Angewandte Informatik, Semester 5*

*Testing*

## Einführung Scrum

*PRODUCT BACKLOG - Ist eine Liste an Anforderungen für das Gesamte Projekt. Dies wird am Anfang des Projektes festgelegt und im laufe der Zeit kontinuierlich erweitert. Die Anforderungen werden nach Notwendigkeit, wirtschaftlichem Nutzen und Risiko priorisiert. Anforderungen im Product Backlog sollen nicht technisch sondern Anwenderorietiert sein.*

*SPRINT BACKLOG - Ähnlich wie der Product Backlog ist der Sprint Backlog ein aktueller Plan des nächsten Sprints. Während eines Sprintes werden die Anforderungen nach Erledigung laufend aktualisiert. Häufig wird der Sprint Backlog als "Taskboard" benutzt.*

*SPRINT - Ist ein Arbeitsabschnitt, in welchem ein bestimmter neuer Teil eines Projektes implementiert werden soll. Sprints sollten zwischen 2 und 4 Wochen angesetzt sein und es sind keine Änderungen an einem laufendem Sprint vorzunehmen, die das Sprintziel beeinflussen. Generell sollte das Ergebnis des Sprintes ein laufendes Programmstück sein, sollte dieses Ziel aus den verschiedensten Gründen nicht erreicht werden können, so wird der Sprint abgebrochen und ein neuer mit angepassten Umfang erstellt.*

*DAILY SCRUM MEETING - Wie der Name schon sagt, ein tägliches Meeting des Entwicklerteams, dass dazu dient jeden auf den neuesten Stand zu bringen und zu synchronisieren. Das daily meeting sollte nicht länger als 15 Minuten dauern.*

*SPRINT REVIEW MEETING - Das Endmeeting eines Sprints indem das Progammstück welches im Sprint erstellt wurde präsentiert, demonstriert und vom Product Owner abgenommen wird.*

*Scrum arbeitet im wesentlichen mit 6 Prinzipien:*

*Transparenz*

*Beobachten und Anpassen*

*Time-Boxing*

*Dinge werden abgeschlossen*

*Maximierung vom Geschäftswert*

*Teams scheitern nicht*

Einführung Android Programmierung:

*Android Apps werden nativ in Java Programmiert.*

*Wir brauchen eine Entwicklungsumgebung (IntelliJ, Eclipse, Android Studio,…). Zusätzlich um Java Code zu kompilieren brauchen wir das Java JDK. Und um die Android Funktionen einzubinden, benötigen wir das Android SDK.*

*Nachdem alles installiert wurde, findet man im Ordner des Android SDK den SDK Manager. Dieser dient dazu Funktionen für die einzelnen Android Versionen und etwaige Extra Libraries runter zu laden.*

*Am Wichtigsten hierbei ist die höchste Android Version für die man programmieren will. Falls man Google Dienste einbinden möchte, muss man die Google Play Dienste ebenfalls auswählen. Alle Features können jederzeit über den SDK Manager heruntergeladen werden.*

*Anschließend müssen wir die ADT (Android Development Tools) in seiner Entwicklungsumgebung einbinden.*

*Im Android AVD Manager kann man Emulatoren erstellen, auf denen man seine Applikation testen kann. Jedoch kann man dies auch auf seinem eigenen Handy testen (wenn man dieses entsprechend präpariert).*

*Die eigentliche Programmierung läuft etwas anders ab als in Java. Android verbindet .xml Dateien mit .java Dateien. Die .xml Dateien dienen hierbei als grundsätzliche Struktur, welche man in einer graphischen Oberfläche verändern kann. Im .java File kann man seinen Oberflächen Funktionalität zuweisen.*

*KUNDENFRAGEN*

*Welche Android Versionen sollen unterstützt sein (von/bis)?*

*Was ist dem Kunden bei dieser App besonders wichtig?*

*Welche Devices? (Tablet, Handy etc.)*

*Minimum response time? (Also wie schnell soll die App sein)*

*Welche Sprachen sollen unterstützt werden? (Deutsch, Englisch, andere)*

*Wie soll der Multiplayer realisiert werden? Bluetooth, WiFi?*

*Was soll bei einem möglichen Verbindungsabbruch passieren?*

*Können Libraries/APIs verwendet werden?*

*Gibt es Designvorschriften (evtl. Blöcke in gewissen Farben oder ähnliches)? Soll der User selbst Farben auswählen können oder spielt das keine Rolle?*

*Sollen die Benutzer immer neu generiert werden oder soll man einen Spieler erstellen können, mit welchem man sich später wieder einloggen kann? Gibt es Achievements, eine Freundesrangliste etc.?*

*Sollen Daten persistent gemacht werden? (Spielstände, Statisiken von Spielern usw.)*

*Sollen die User miteinander kommunizieren können? Wie?*

*Soll ein Spiel auch mit zwischenzeitlichen Programmschließungen funktionieren?* Also *zeitverschoben? (Wäre bei diesem rundenbasierten Spiel ja möglich)*

*Sollen die Einstellungen fix sein oder soll es dem User ermöglicht werden, Parameter selbst zu wählen?*

*Zielgruppe für die Applikation?*

*Wie sehen die Vorstellungen für die grafische Umsetzung aus?*

*Fragen bezüglich Schummelfunktion, Sprachsteuerung, Gestiken (ob die ausgedachten Ideen umgesetzt werden können)*

*Spezialeffekte? Verschiedene Spielmodi?*

*Was bedeutet "Erfolg" für das Projekt?*

*Gibt es noch weitere Kundengespräche, wenn die Entwicklung fortgeschrittener ist?*

KUNDENANGEBOT

Prof. Martin Pinzger

Universitätsstraße 65-67

9020 Klagenfurt am Wörthersee

Auftrag vom 05.05.2016

Unser Angebot "Brettspiel Blokus"

Sehr geehrter Herr Prof. Martin Pinzger

Herzlichen Dank für ihren Auftrag, für Sie eine Spiele-App für Mobile Devices zu programmieren.

Wie bereits besprochen, haben wir uns für das rundenbasierte Strategie-Brettspiel „Blokus“ entschieden, jedoch in minimal abgeänderter Form – das Ergebnis ist schließlich „Bloxx“.

„Bloxx“ eignet sich besonders für Ihre gestellten Anforderungen, da es einerseits sehr anspruchsvoll, aber auch sehr lustig ist. „Bloxx“ kann entweder zu zweit, aber auch zu dritt oder auch zu viert gespielt werden. Die Spieler matchen sich auf einem 20x20 Feld und versuchen durch strategisch und taktisch kluge Züge so viele Spielsteine wie möglich zu legen und den Gegner schließlich zu Fall zu bringen.

Startet ein Spieler die App, so muss er sich Authentifizieren (Nutzername) und kann nun sofort mit der Auswahl eines bereits bestehenden Spiels oder mit dem Erstellen eines eigenen beginnen. Die Kommunikation zwischen den Geräten wird über ein lokales Netzwerk (WLAN) realisiert.

Die Abänderungen der Originalversion betreffen hauptsächlich eine Schummelfunktion, welche den Spielspaß maßgeblich erhöhen wird. Um das Schummeln und somit einen Vorteil für den/die Gegenspieler abzuwenden, muss man natürlich stets wachsam sein!

Die Interaktion mit dem Smartphone ist außerdem vielseitig – neben den herkömmlichen Touch-Events ist auch eine Schüttelfunktion für die Rundenbeendigung und eine Sprachfunktion für das „Entlarven“ eines Schummlers geplant.

Damit die benutzerspezifischen Präferenzen, wie zum Beispiel die Wahl der Farbe der eigenen Spielsteine oder überhaupt die Menüsprache, nicht zu kurz kommen, wird es natürlich eine Möglichkeit geben, diese selbst einzustellen.  
  
Da unsere Kunden dieses Spiel nicht nur am Smartphone genießen sollen können, wird es selbstverständlich auch eine Version für das Tablet geben. Generell ist „Bloxx“ für Androidversionen ab 4.4.1 (KitKat) konzipiert.

Für die Entwicklung von „Bloxx“ wird, da die Projektgruppe 600 Entwicklungsstunden geschätzt hat, ein Preis von 10.000,00€ veranschlagt. Sollten sich noch Möglichkeiten ergeben, zusätzliche (nicht besprochene) nützliche Funktionalitäten hinzuzufügen, werden wir Sie umgehend informieren – diese wirken sich dann natürlich auch auf den Preis aus. Natürlich können Sie sich bei Fragen umgehend an uns wenden.

Wir verbleiben mit freundlichen Grüßen  
 Gruppe E

*Product-Backlog und Aufwandsschätzung*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Id* | *Beschreibung* | *Typ* | *Story Points* | *Priorität* | *Notizen* |
| *1* | *Der User soll aus mehreren Steinen auswählen können* | *User Story* | *5* | *200* |  |
| *2* | *Der User soll Spielsteine legen können* | *User Story* | *20* | *200* |  |
| *3* | *Die grundlegende GUI* | *Constraint* | *20* | *200* |  |
| *4* | *Der User soll Spielsteine drehen können* | *User Story* | *2* | *200* |  |
| *5* | *Der User soll ein neues Spiel starten können* | *User Story* | *8* | *200* |  |
| *6* | *Der User soll einem bestehendem Spiel beitreten können* | *User Story* | *5* | *200* |  |
| *7* | *Netzwerkkommunikation* | *Constraint* | *20* | *200* |  |
| *8* | *Der User soll schummeln können* | *User Story* | *8* | *180* | *nA* |
| *9* | *Der User soll andere Schummler ertappen können* | *User Story* | *2* | *150* |  |
| *10* | *Das System soll schnell auf die Eingaben des Users reagieren* | *Constraint* | *2* | *150* |  |
| *11* | *Der User soll jederzeit wissen, wieviele Quadrate er belegt* | *User Story* | *1* | *100* |  |
| *12* | *Das System soll sinnvolle Fehlermeldungen anzeigen* | *Constraint* | *5* | *100* |  |
| 13 | Ansprechende GUI | Constraint | 8 | 100 |  |
| 14 | Am Ende des Spiels sollte eine Spielstatistik angezeigt werden | User Story | 3 | 40 |  |
| 15 | Der User soll Einstellungen tätigen können (Farbe der Steine,...) | User Story | 2 | 20 |  |
| 16 | Der User kann zwischen Englisch und Deutsch umstellen | User Story | 2 | 10 |  |
| 17 | Der User soll ein Konto anlegen können (Für Highscores,…) | User Story | 13 | 10 |  |
| 18 | Der User soll sich anmelden können (nur Name) | User Story | 2 | 10 |  |
|  |  |  | 128 |  |  |

SPRINTS

*Sprint 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *1* | *Der User soll Spielsteine drehen können* | *User Story* | *2* | *200* |
| *2* | *Der User soll ein neues Spiel starten können* | *User Story* | *8* | *200* |
| *3* | *Der User soll einem bestehendem Spiel beitreten können* | *User Story* | *5* | *200* |
| *4* | *Netzwerkkommunikation* | *Constraint* | *20* | *200* |
| *5* | *Der User soll schummeln können* | *User Story* | *8* | *180* |
|  |  |  |  |  |

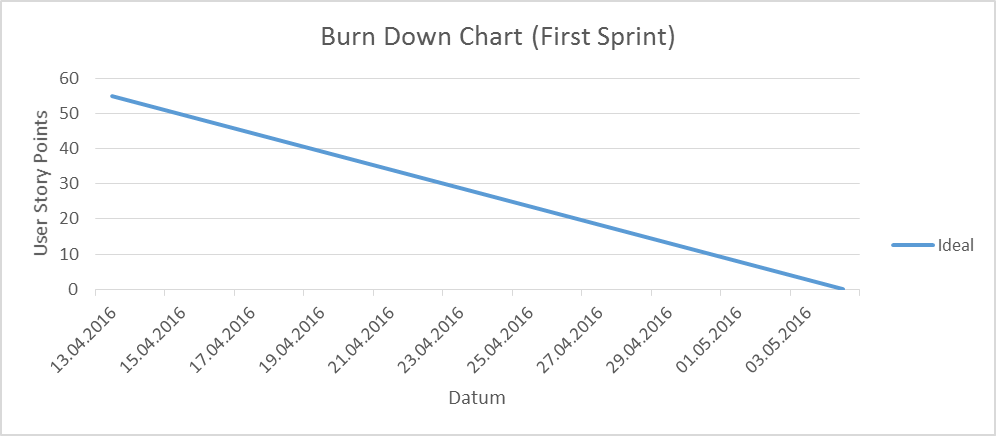
*Sprint 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *6* | *Der User soll aus mehreren Steinen auswählen können* | *User Story* | *5* | *200* |
| *7* | *Der User soll Spielsteine legen können* | *User Story* | *20* | *200* |
| *8* | *Die grundlegende GUI* | *Constraint* | *20* | *200* |
| *9* | *Netzwerkkommunikation* | *Constraint* | *20* | *200* |

*Sprint 3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *10* | *Der User soll andere Schummler ertappen können* | *User Story* | *2* | *150* |
| *11* | *Das System soll schnell auf die Eingaben des Users reagieren* | *Constraint* | *2* | *150* |
| *12* | *Der User soll jederzeit wissen, wieviele Quadrate er belegt* | *User Story* | *1* | *100* |
| *13* | *Das System soll sinnvolle Fehlermeldungen anzeigen* | *Constraint* | *5* | *100* |
| *14* | *Ansprechende GUI* | *Constraint* | *8* | *100* |
| *15* | *Am Ende des Spiels sollte eine Spielstatistik angezeigt werden* | *User Story* | *3* | *40* |
| *16* | *Der User soll Einstellungen tätigen können (Farbe der Steine,...)* | *User Story* | *2* | *20* |
| *17* | *Der User kann zwischen Englisch und Deutsch umstellen* | *User Story* | *2* | *10* |
| *18* | *Der User soll ein Konto anlegen können (Für Highscores,…)* | *User Story* | *13* | *10* |
| *19* | *Der User soll sich anmelden können (nur Name)* | *User Story* | *2* | *10* |

BURNDOWN CHARTS



 STUNDENLISTE

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | STUNDEN | TOBIAS |  | MARKUS |  | MANUEL |  | DANIEL |  | LAURA |
| 07. Mrz | 2 | UEBLATT | 2 | UEBLATT | 2 | UEBLATT | 2 | UEBLATT | 2 | UEBLATT |
| 08. Mrz |  |  |  |  |  |  |  |  |  |  |
| 04. Apr | 2 | UEBLATT | 2 | UEBLATT | 2 | UEBLATT | 2 | UEBLATT | 2 | UEBLATT |
| 05. Apr |  |  |  |  |  |  |  |  |  |  |
| 06. Apr |  |  |  |  |  |  |  |  |  |  |
| 07. Apr |  |  |  |  |  |  |  |  |  |  |
| 08. Apr |  |  |  |  |  |  |  |  |  |  |
| 09. Apr |  |  |  |  |  |  |  |  |  |  |
| 10. Apr |  |  | 2 | Design der Spielelemente |  |  |  |  |  |  |
| 11. Apr |  |  |  |  |  |  |  |  |  |  |
| 12. Apr |  |  |  |  |  |  | 2 | Installation und Einarbeitung Android Studio |  |  |
| 13. Apr |  |  |  |  |  |  |  |  |  |  |
| 14. Apr |  |  | 2 | Einlesen |  |  |  |  |  |  |
| 15. Apr | 2,00 | Einarbeiten Android Studio | 3 | Einlesen |  |  |  |  |  |  |
| 16. Apr |  |  | 4 | Einlesen /Activity-Change |  |  |  |  | 2 | Einarbeiten |
| 17. Apr | 2,00 | Einarbeiten Android Studio, Netzwerkkommunikation | 3 | Hauptmenü | 2 | Einarbeiten Android Studio |  |  |  |  |
| 18. Apr |  |  | 2 | Einlesen in Layouts/Views | 2 | Tutorials:Android Studio | 2 | Troubleshooting Android Studio | 2 | Tutorials |
| 19. Apr |  |  | 1 | Einlesen in Layouts/Views |  |  | 1 | Tutorials Android Studio und Sampleprojects |  |  |
| 20. Apr |  |  | 3 | Einlesen in Layouts/Views & planung von Klassen-/Datenstrukturen |  |  |  |  |  |  |
| 21. Apr |  |  | 5 | implementierung Singleplayer / GameLogic |  |  |  |  |  |  |
| 22. Apr |  |  | 4 | implementierung Singleplayer / GameLogic |  |  |  |  |  |  |
| 23. Apr | 2 | Netzwerkkommunikation | 8 | implementierung Singleplayer |  |  |  |  |  |  |
| 24. Apr |  |  | 3 | implementierung Singleplayer /Buffixing | 3 | Einrichten von Android Umgebung an Hauptrechner + MockUp von Startbildschirm |  |  |  |  |
| 25. Apr | 4 | Netzwerkkommunikation | 2 | implementierung von einzelnen Multiplayer-Funktionen |  |  |  |  |  |  |
| 26. Apr | 3 | Netzwerkkommunikation | 1 | implementierung von einzelnen Multiplayer-Funktionen | 4 | Verknüpfen von Startbildschirm und Spielentwurf + Aktivitätsübegreifender Funktion |  |  | 3 | Projektmappe |
| 27. Apr |  |  |  |  |  |  | 2 | Einarbeiten in vorhandenen Programmcode |  |  |
| 28. Apr |  |  |  |  |  |  |  |  |  |  |
| 29. Apr |  |  |  |  |  |  |  |  |  |  |
| 30. Apr | 2 | Netzwerkkommunikation Google Nearby connections einbetten |  |  |  |  |  |  |  |  |
| 01. Mai | 4 | Netzwerkkommunikation anpassen |  |  |  |  |  |  |  |  |
| 02. Mai | 4 | Netzwerkkommunikation anpassen |  |  | 5 | Steine können nicht mehr über den Rand platziert werden + Troubleshooting (Steine entfernen) |  |  |  |  |
| 03. Mai | 6 | Netzwerkkommunikation anpassen |  |  | 3,50 | Steine werden nicht mehr überschrieben (noch nicht perfekt) + alle Steine ohne Exceptions |  |  |  |  |
| 04. Mai |  |  |  |  |  |  |  |  | 2 | Projektmappe |
| 05. Mai |  |  |  |  |  |  |  |  |  |  |

,,,,.