Gruppe P24  
Matthias Ridder, Dominic von Zielinski

Domino

Projektdokumentation

Inhaltsverzeichnis

[1. Pflichtenheft 2](#_Toc422945807)

[Pflichtenheft 2](#_Toc422945808)

[1 Management- und Dokumentationsattribute 2](#_Toc422945809)

[2 Visionen und Ziele 2](#_Toc422945810)

[3 Rahmenbedingungen 3](#_Toc422945811)

[4 Kontext und Überblick 4](#_Toc422945812)

[5 Funktionale Anforderungen 4](#_Toc422945813)

[6 Qualitätsanforderungen 4](#_Toc422945814)

[7 Abnahmekriterien 5](#_Toc422945815)

[8 Quellen 5](#_Toc422945816)

[9 Hinweis zu dieser Vorlage 5](#_Toc422945817)

[9.1 Literaturliste 5](#_Toc422945818)

[2. Projektmanagement 6](#_Toc422945819)

[2.1 Rollen 6](#_Toc422945820)

[3. Projektplanung 6](#_Toc422945821)

[4. UML-Diagramme 7](#_Toc422945822)

[4.2 Klassendiagramm 7](#_Toc422945823)

# Pflichtenheft

Pflichtenheft

*Domino*

*P24*

*24.06.2015*

# Management- und Dokumentationsattribute

|  |  |
| --- | --- |
| Dokumentationsattribute | |
| Autor | Dominic von Zielinski, Matthias Ridder |
| Eindeutige Teamnummer | P24 |
| Quelle |  |
| Version | 1.0 |
| Bearbeitungsstatus |  |

# Visionen und Ziele

/PV10/ **Mehrspielermodus**

Standardmäßig spielen entweder zwei menschliche Spieler/ ein menschlicher Spieler und ein Computerspieler gegeneinander.

Die Vision des Mehrspielermodus soll das Ganze auf eine Spieleranzahl von 3/4 bringen. Die Erweiterung der Spieleranzahl gibt einer Gruppe die meist aus mehr als 2 Menschen besteht die Möglichkeit sich gleichzeitig in einer Runde zu messen

/PV20/ **KI**

Die Realisierung einer künstlichen Intelligenz ist eine weitere Vision für unser Programm. Die künstliche Intelligenz soll das Spielerlebnis des Spielers verbessern und ihm das Gefühl geben gegen einen richtigen Spieler zu spielen.

Neben dem üblichen Regelwerk muss die KI folgendes wissen:

1. Welche Steine sie auf der Hand hat
2. Welche Steine sie an welche Steine anlegen kann
3. Wie die Punkte für eine Runde und die Maximalpunktzahl errechnet wird
4. Sie muss die Sonderregeln kennen
5. Sie muss wissen, dass sie nach einer verlorenen Runde ihre restlichen Steine zusammenzählt
6. Sie muss wissen, dass sie Steine vom Talon nehmen muss wenn sie keine passenden Steine auf der Hand hat
7. Sie muss wissen wann sie gewonnen hat
8. Sie muss wissen wer das Spiel beginnt/den ersten Zug machen darf

/PV30/ **Verschiedene Schwierigkeitsmodi**

Zusätzlich zur KI gehört die Implementierung von verschiedenen Schwierigkeitsmodi ebenfalls zu unseren Visionen. Angesetzt ist fürs erste eine feste Schwierigkeitsstufe. Bei Zeiten kann dies auf bis zu 3 (leicht, mittel und schwer) erweitert werden.

Sinn der verschiedenen Schwierigkeitsmodi ist ein abwechslungsreicheres Spielerlebnis. Dem einen fällt es leichter/macht es mehr Spaß gegen einen schwierigen Computergegner zu spielen während der andere lieber auf ein leichtes Schwierigkeitslevel zurückgreifen will, da er noch nicht so erfahren ist.

/PZ10/ **Aufbau/Implementierung der Logik**

Die Implementierung der Logik ist das Herzstück des Spiels. Die Logik beinhaltet die

Korrekte Handhabung der Spielregeln, den in einer korrekten Reihenfolge ablaufenden Spielablauf und alle relevanten Informationen die einen korrekten und vollständig richtigen Spielverlauf beeinflussen.

/PZ20/ **GUI**

Ebenso wichtig wie die Logik ist die GUI. Die GUI bietet dem Spieler eine visuelle Plattform auf der er den Verlauf des Spiels, die Punkte, die Steine und die Züge in visueller Form aufnehmen und verarbeiten kann. Außerdem vereinfacht die GUI dem Spieler das spielen. Er kann sich ganz auf das Spiel konzentrieren und wird nicht von diversen Störfaktoren abgelenkt.

Die GUI soll folgende Punkte beinhalten:

1. Die Darstellung des Spielfelds in Form eines Fensters
2. Die Darstellung der 28 Spielsteine (Eigene Hand, gegnerische Hand, Talon und Spielfeld)
3. Die Darstellung der Punkte (Punktezähler)
4. Die Aufteilung des Spielfelds (z.B. Anzeige von eigenen Steinen getrennt von denen des Gegners) um dem Spieler eine klare Übersicht des Spielfelds zu gewährleisten.

# Rahmenbedingungen

/PR10/ **Anwendungsbereich**

Videospiel

/PR20/ **Zielgruppe**

Menschen zwischen 6-99 Jahren, die Gesellschaftsspiele mögen

/PR30/ **Betriebsbedingungen**

Hardware: PC, Bildschirm mit einer Mindestauflösung von 1600x900

Software: Windows und mindestens Java Version 7

Schnittstellen: Maus

# Kontext und Überblick

/PK10/ **Windows und Java**

Das Spiel wird auf jedem javafähigen Windows-PC, Mac oder Linux-System funktionieren. Optimiert wird es jedoch für Windows-Systeme.

Die eingesetzte Java-Version ist Version 7

/PÜ10/ **Domino**

Programmiert wird das Legespiel Domino. Es wird nach den Regeln des Muggins bzw. All-Fives Spiels gespielt.

# Funktionale Anforderungen

/PF10/ **Intuitive Bedienung mit Maus**

/PF20/ **Leicht zu lernen, schwer zu meistern**

/PF30/ **Flüssiger Spielablauf**

/PF40/ **Fehlerfreies Spiel**

# Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Systemqualität | Sehr gut | Gut | Normal | Nicht relevant |
| Funktionalität |  | X |  |  |
| Zuverlässigkeit | X |  |  |  |
| Benutzbarkeit |  | X |  |  |
| Effizienz |  |  |  | X |
| Wartbarkeit |  |  | X |  |
| Portabilität |  |  |  | X |

Tabelle 1: Qualitätsanforderungen

/PQF10/ **Funktionalität**

Das Spiel muss flüssig funktionieren, ohne dass es abstürzt.

/PQZ10/ **Zuverlässigkeit**

Zuverlässigkeit der Logik. Diese darf keine falschen Spielzüge zulassen und muss die Punkte immer richtig berechnen.

/PQB10/ **Benutzbarkeit**

Intuitive Benutzbarkeit und leicht verständliche GUI.

/PQE10/ **Effizienz**

Da es sich um ein kleines Spiel handelt, ist die Effizienz des Programms nicht relevant.

/PQW10/ **Wartbarkeit**

Fehlerhafter Code (Bugs) sollen im Nachhinein beseitigt werden können. Die Funktionalität des Quellcodes soll jedoch im Vordergrund stehen.

Der Quellcode kann im Nachhinein übersichtlicher gestaltet werden.

/PQP10/ **Portabilität**

Die Portabilität des Programms ist in keiner Hinsicht relevant.

# Abnahmekriterien

* Funktionalität der Spielmechanik
* Keine deutlich sichtbaren Fehler
* Intuitive Bedienbarkeit

# Quellen

* <http://docs.oracle.com/javase/7/docs/api/overview-summary.html>
* <https://docs.oracle.com/javase/tutorial/2d/images/index.html>

# Hinweis zu dieser Vorlage

Die Vorlage für dieses Pflichtenheft wurde Balzert (2009), S. 492 ff. entnommen.

## Literaturliste

Balzert, Helmut (2009). Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. 3. Auflage. Heidelberg: Spektrum, Seite 492 ff.

# Projektmanagement

## Rollen

Die Teammitglieder übernahmen folgende Rollen:

Matthias Ridder: Dokumentation, Softwareentwicklung, Tester

Dominic von Zielinski: Dokumentation, Softwareentwicklung, Tester

# Projektplanung

# UML-Diagramme

## Klassendiagramm

