Software Requirements Specification

Version 2.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 21.10.2016 | 1.0 | First SRS Document | André, Luka, Katharina |
| 04.04.2017 | 2.0 | Complete New SRS Document | André, Lukas, Katharina |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Overall Description 3

3. Specific Requirements 3

3.1 Functionality 3

3.1.1 <Functional Requirement One> 3

3.2 Usability 3

3.2.1 <Usability Requirement One> 3

3.3 Reliability 3

3.3.1 <Reliability Requirement One> 3

3.4 Performance 3

3.4.1 <Performance Requirement One> 3

3.5 Supportability 3

3.5.1 <Supportability Requirement One> 3

3.6 Design Constraints 3

3.6.1 <Design Constraint One> 3

3.7 On-line User Documentation and Help System Requirements 3

3.8 Purchased Components 3

3.9 Interfaces 3

3.9.1 User Interfaces 3

3.9.2 Hardware Interfaces 3

3.9.3 Software Interfaces 3

3.9.4 Communications Interfaces 3

3.10 Licensing Requirements 3

3.11 Legal, Copyright, and Other Notices 3

3.12 Applicable Standards 3

4. Supporting Information 3

Software Requirements Specification

# Introduction

Unser Ziel ist es, das legendäre Tetris Spiel aus den 80ern wieder aufleben zu lassen. Hier werden einige Änderungen vorgenommen und verschiedene, noch nie dagewesene, Spielmodi hinzugefügt.

## Purpose

Dieses Software Requirement Specification Dokument soll einen Überblick über die Anforderungen und Dokumente unseres Projektes bieten.

## Scope

Dieses Software Requirement Specification Dokument dient als Richtlinie für die Entwickler sowie als Überblick für externe.

## Definitions, Acronyms, and Abbreviations

AnKaLu = **An**dré, **Ka**tharina, **Lu**ka

n/a = not applicable

## Document - References

|  |  |
| --- | --- |
| Blog | <https://ankalublog.wordpress.com/blog/> |
| Github | <https://github.com/AlSalad/Ankalu> |
| IntelliJ | <https://www.jetbrains.com/idea/> |
| Jira | <http://193.196.7.27:8080/secure/Dashboard.jspa> |
| Use-Case: Musik ändern |  |
| Use-Case: Standard-Mode spielen |  |
| Use-Case: Fast-Mode spielen |  |
| Use-Case: Reverse-Mode spielen |  |
| Use-Case: Epilepsie-Mode spielen |  |

# Overall Description

Es handelt sich um eine Desktopanwendung, die das legendäre Tetris mit einigen speziellen Features wieder aufleben lässt. Die Grundstruktur stellt der Standard-Mode dar, der dem Standard Tetris entspricht. Zusätzlich kann man individuelle Spieleinstellungen, wie z.B. Musik ändern oder Design ändern, im Menü einstellen. Des Weiteren stehen noch nie dagewesene Spielmodi zur Verfügung:

* Epilepsie-Mode: Der Hintergrund flackert in verschiedenen Farben.
* Fast-Mode: Die Blöcke bewegen sich schneller als zuvor Richtung Boden.
* Reverse-Mode: Die Steuerung der Blöcke ist spiegelverkehrt.

## Product Perspective

Dieses Spiel soll den Tetrisfans die Möglichkeit bieten, das damalige Spieleerlebnis wieder aufleben zu lassen. Zusätzlich sorgen die verschiedenen Spielmodi für eine frische Abwechslung, da das Standard Tetris, vorallem für junge Leute, sehr schnell eintönig werden kann.

## Product Functions

* Standard Tetris spielen
* Epilepsie Tetris spielen
* Fast Tetris spielen
* Reverse Tetris spielen
* Musik ändern und hinzufügen

## User Characteristics

Der Nutzer wird durch den Highscore motiviert weiter zu spielen und dadurch seinen Highscore zu verbessern. Durch die verschiedenen Modi wird es dem Nutzer nicht langweilig und kann sich auch mit anderen in diesem Spiel messen.

## Contraints

Da diese Datei später als Anwendung (.exe) vorliegt, benötigt man nichts weiter als einen funktionierenden PC, mit einer funktionierenden Maus (damit die Anwendung gestartet und im Menü die Modi etc. ausgewählt werden können) und eine funktionierende Tastatur, damit die einzelnen Blöcke gesteuert werden können.

## Assumptions and Dependencies

* IDE: IntelliJ
* Versionskontrolle: GitHub
* Scrum: Jira
* Programmiersprache: Java
* Tests: Cucumber, Feature File Test

# Specific Requirements

[This section of the **SRS** contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the Use Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

## Functionality

[This section describes the functional requirements of the system for those requirements that are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods may also be appropriate; for example, organization by user or organization by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, and the like, are employed to capture the functionality, this section of the document would refer to the availability of that data, indicating the location and name of the tool used to capture the data.]

### <Functional Requirement One>

[The requirement description.]

## Usability

[This section includes all those requirements that affect usability. For example,

* specify the required training time for a normal users and a power user to become productive at particular operations
* specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like
* specify requirement to conform to common usability standards, such as IBM’s CUA standards Microsoft’s GUI standards]

### <Usability Requirement One>

[The requirement description goes here.]

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

* Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.
* Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.
* Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?
* Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.
* Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).
* Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

* Response time for a transaction (average, maximum)
* Throughput, for example, transactions per second
* Capacity, for example, the number of customers or transactions the system can accommodate
* Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
* Resource utilization, such as memory, disk, communications, and so forth.

### <Performance Requirement One>

[The requirement description goes here.]

## Supportability

n/a

## Design Constraints

Es handelt sich Desktopanwendung. Die Architektur und das Klassendiagramm können aus dem „Sotware Architecture Document“ entnommen werden.

## On-line User Documentation and Help System Requirements

Das Menü des Spieles und die einzelnen Modi sind selbsterklärend. Lediglich für das verschiebender Blöcke wird eine kleine Hilfeseite zur Verfügung stehen.

## Purchased Components

n/a

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

n/a

### Software Interfaces

n/a

### Communications Interfaces

n/a

## Licensing Requirements

n/a

## Legal, Copyright, and Other Notices

n/a

## Applicable Standards

n/a

# Supporting Information

n/a