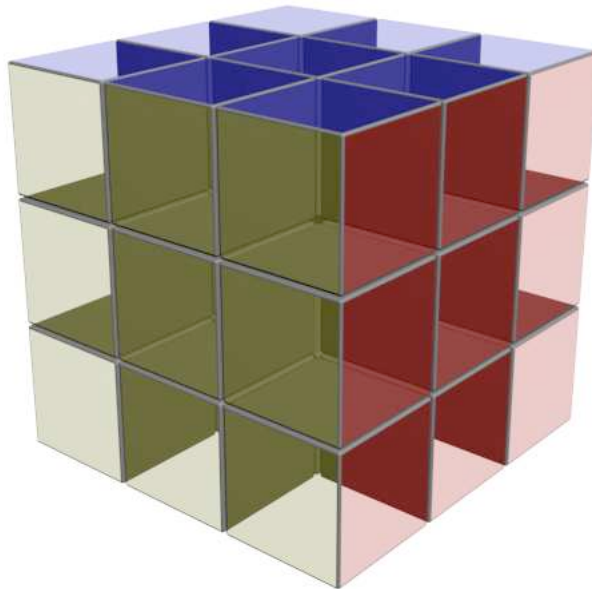




»Pflichtenheft« v 1.0

$$YaRC$$

Yet another Rubik Cube



Phase	Phasenverantwortliche(r)	E-Mail

Inhaltsverzeichnis

1	Einleitung	3
2	Zielbestimmungen	4
2.1	Musskriterien	4
2.2	Wunschkriterien	4
2.3	Abgrenzungskriterien	5
3	Produkt-Einsatz	6
3.1	Anwendungsbereich	6
3.2	Zielgruppe	6
3.3	Betriebsbedingungen	6
4	Produktumgebung	7
4.1	Software	7
4.2	Hardware	7
5	Produktfunktionen	8
5.1	Grundfunktionen	8
5.2	Erweiterte Funktionen	8
6	Produkt-Daten	9
6.1	System-Daten	9
6.2	Benutzer-Daten	9
7	Systemmodell	10
8	Produkt-Leistungen	11
9	Benutzeroberfläche	12
9.1	Einführung	12
9.2	Startmenü	12
9.2.1	Nur Grundfunktionen	12
9.2.2	Mit erweiterten Funktionen	13
9.3	Spielmenü	13
9.3.1	Nur Grundfunktionen	13
9.3.2	Mit erweiterten Funktionen	14
9.4	Screenshots	15
10	Qualitätszielbestimmungen	17
11	Testfälle und Testszenarien	18
11.1	Testfälle	18
11.2	Testszenarien	19
12	Entwicklungs-Umgebung	24
12.1	Software	24
12.2	Hardware	24
	Glossar	25

1 Einleitung

Das Ziel dieses Projekts ist, ein Handyspiel zu entwickeln, mit dem ein Spieler einen Rubik-Würfel lösen kann. Der Rubik-Würfel, auch Zauberwürfel genannt, ist ein mechanisches Geduldsspiel, das 1975 von dem Ungarn Ernő Rubik erfunden wurde.

Es handelt sich dabei um einen Würfel, der in Höhe, Breite und Tiefe in drei Ebenen unterteilt ist, die sich durch 90-Grad-Drehungen um ihre jeweilige Raumachse zur Deckung bringen lassen. Dadurch können Position und Lage der insgesamt 26 Teilwürfel fast beliebig geändert werden. Auf die nach außen sichtbaren Flächen der Steine sind kleine Farbflächen geklebt. In der Grundstellung sind die Steine so geordnet, dass jede Seite des Würfels eine einheitliche, aber von Seite zu Seite unterschiedliche Farbe besitzt.

Ziel des Spiels ist es, den Würfel wieder in seine Grundstellung zu bringen, nachdem er zuvor durch mehrfaches Verdrehen in eine zufällige Ausgangsposition gedreht wurde. Auf den ersten Blick erscheint diese Aufgabe außerordentlich schwierig, jedoch wurden schon frühzeitig Strategien entwickelt, deren Kenntnis ein relativ leichtes Lösen gestattet.

Er erfreute sich insbesondere Anfang der 1980er Jahre bei Groß und Klein großer Beliebtheit und war in vielen Haushalten zu finden.

Lust auf ein Spiel mit dem Rubik-Würfel bekommen? Testen Sie unser Programm und Sie werden begeistert sein!

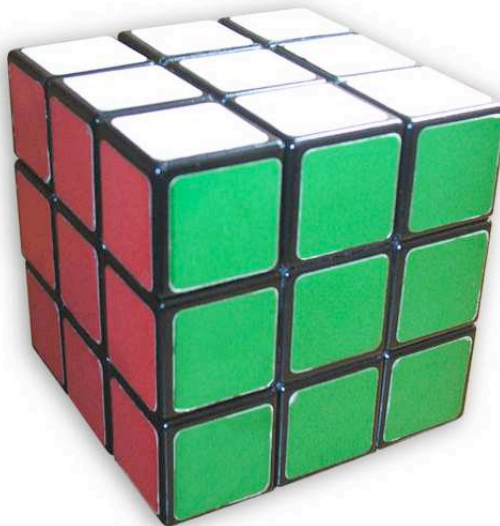


Abbildung 1 - Rubik-Würfel

2 Zielbestimmungen

2.1 Musskriterien

- Das Programm läuft in der Simulationsumgebung [KToolbar](#) und auf den meisten Java-fähigen Handys.
- Nachdem das Programm geladen wurde, wird eine menügesteuerte Benutzerführung (Startmenü) angezeigt.
- Nachdem der Benutzer ein neues Spiel gestartet hat, erstellt das Programm eine neue Spielsituation mit einem zufälligen Würfel.
- Der Würfel wird graphisch mittels einer [GUI](#) dargestellt. Der Benutzer kann den Würfel von allen Seiten betrachten.
- Das Erstellen einer neuen Spielsituation und das Lösen des Würfels erfolgt jeweils mit einem [Lösungsalgorithmus](#).
- Der Benutzer kann den Rubik-Würfel manuell mit Hilfe der Handy-Tasten lösen. Dabei können die verschiedenen Ebenen des Würfels beliebig gegeneinander gedreht werden.
- Durch eine Hilfe-Funktion wird für den Benutzer der nächste Schritt zur Lösung ausgeführt.
- Durch beliebig häufiges Betätigen der Hilfe-Funktion wird das Spiel richtig gelöst (halbautomatisches Lösen des Würfels).
- Der Benutzer kann durch die [undo](#) -Funktion die letzten drei Schritte rückgängig machen.
- Durch eine [redo](#) -Funktion kann ein rückgängig gemachter Schritt wieder hergestellt werden.
- Der Rubik-Würfel kann vollautomatisch gelöst werden.
- Das Programm kann als Bildschirmschoner verwendet werden, bei dem immer wieder ein neuer Würfel erzeugt und automatisch gelöst wird.
- Dem Benutzer steht eine Spielanleitung und eine genaue Beschreibung der Tastenbelegungen zur Verfügung. Letztere kann er jederzeit über ein Menü aufrufen.

2.2 Wunschkriterien

- Das Programm soll über eine [Bestenliste](#) verfügen, in der die erreichten Punkte zusammen mit dem Namen gespeichert sind. Dabei wird der Benutzer automatisch, nachdem er einen [Highscore](#) - Tabellenpatz erzielt hat, aufgefordert, seinen Namen einzugeben und zu speichern. Danach wird die neue Bestenliste angezeigt. Auf der Liste erscheinen höchstens fünf Einträge. Die Punkte werden abhängig von der benötigten Zeit und vom Schwierigkeitsgrad berechnet.
- Das Programm soll über eine Möglichkeit verfügen, den aktuellen "Spielstand" zu speichern, um zu einem späteren Zeitpunkt wieder weiterspielen zu können. Dabei wird auch die bisher benötigte Zeit gesichert.
- Der Benutzer kann den Schwierigkeitsgrad im Menü wählen. Es gibt drei Stufen: Anfänger, Fortgeschrittene und Profis.
- Der Benutzer kann zwischen zwei Oberflächenarten des Würfels im Menü auswählen (entweder Farbe oder [Textur](#)).
- Wenn ein Würfel gelöst wurde, wird eine Melodie gespielt. Wenn der Spieler den [Highscore](#) geknackt hat, soll eine besondere Melodie erklingen.
- Der Benutzer kann die Lautstärke verändern und den Ton deaktivieren.

- Wenn das Spiel z.B. durch einen eingehenden Anruf unterbrochen wird, wird automatisch eine Pause eingelegt (d.h. die Spielzeit wird angehalten). Nach dem Telefonat kann der Spieler das Spiel durch Drücken einer Taste fortsetzen.
- Der Benutzer kann durch einen Button "Pause" das Spiel anhalten. Durch Drücken einer beliebigen Taste wird das Spiel wieder fortgesetzt.
- Mehrsprachige Menüführung

2.3 Abgrenzungskriterien

- Der Benutzer kann nicht ein spezielles Spiel starten, sondern die Spiele werden immer zufällig erzeugt.
- Der Rubik-Würfel ist ein 3x3x3 Würfel. Die Größe des Würfels kann also nicht vorher vom Spieler ausgewählt werden.
- Das Spiel lässt sich nicht direkt (ohne Emulationsschicht) auf anderen Medien (z.B. Computer) spielen.
- Es wird keine Multiplayer- Funktionalität geben. Das Spiel verfügt ausschließlich über einen Singleplayer-Modus.

3 Produkt-Einsatz

3.1 Anwendungsbereich

Das Programm dient der Unterhaltung. Ein Benutzer kann das Spiel auf seinem Handy spielen. Außerdem kann das Programm als Bildschirmschoner verwendet werden, bei dem immer wieder ein Rubik-Würfel erzeugt und anschließend vollautomatisch gelöst wird.

3.2 Zielgruppe

Die Bedienung dieses Spiels ist möglichst einfach gehalten, so dass jeder, der ein modernes Handy bedienen kann, auch mit dem Handyspiel gut umgehen kann. Es sind also keine besonderen Kenntnisse des Benutzers erforderlich. Außerdem gibt es spielbegeisterte Menschen in jeder Altersgruppe, so dass sich auch in dieser Hinsicht die Zielgruppe nicht genau definieren lässt. Auch für Kinder ab dem Grundschulalter ist dieses Spiel gut geeignet. Durch die optionale Verwendung von **Piktogrammen** als **Textur** auf den Würfeln ist es auch rot-grün-blinden Personen möglich das Spiel zu spielen.

3.3 Betriebsbedingungen

Es müssen folgende Voraussetzungen erfüllt sein:

- Java-fähiges, betriebsbereites Handy mit Farbdisplay (Bei Realisierung des Wunschkriteriums "Texturen" genügt ein Graustufen-Display mit passender Auflösung.).
- Auf dem Handy muss ausreichend Speicherplatz (≥ 1 MB) zur Verfügung stehen.
- Während des Spielens muss die Stromversorgung (geladener Akku) gewährleistet sein.
- Sowohl Spiel als auch Bildschirmschoner laufen ohne Zeitbeschränkung und Aufsicht.

4 Produktumgebung

4.1 Software

- [KToolbar](#)
- Die [J2ME](#) -Laufzeitumgebung mit [CLDC](#) 1.1 Konfiguration und [MIDP](#) 2.0 Profil.
- Die Mobile3D Graphics API (JSR 184 Mobile 3D Graphics) für [J2ME](#) .

4.2 Hardware

- Ein [CLDC](#) 1.1 fähiges Java-Handy (Smartphone) mit folgenden Eigenschaften:
 - Farbdisplay (bzw. Graustufen) mit minimaler Auflösung von 96x54 Pixel. (Empfohlen: [QQVGA](#) 160x120 Pixel)
 - Mindestens 32 KB [flüchtiger Speicher](#) für die Ausführung einer [Java Virtual Machine \(JVM\)](#) .
 - Eingabegerät (Ein-/ Zweihandtastatur oder [Touchscreen](#)).

5 Produktfunktionen

5.1 Grundfunktionen

Nr.	Beschreibung	Automatisch/ Benutzer
/F010/	Anzeigen des Begrüßungsbildschirms	A
/F020/	Anzeigen des Startmenüs (bei Programmstart und nach Spielende)	A
/F030/	Anzeigen der Spielanleitung	B
/F040/	Anzeigen der Tastenbelegung	B
/F050/	Spiel starten	A/B
/F060/	Spiel beenden	A/B
/F070/	Erstellen eines zufälligen Rubikwürfels	A
/F080/	Anzeigen des aktuellen Rubikwürfels	A
/F090/	Drehen des Würfels	A/B
/F100/	Auswählen der zu drehenden Ebene und der Drehrichtung	B
/F110/	Drehen jeweils einer Ebene (zum Lösen des Würfels)	A/B
/F120/	Schritte rückgängig machen durch undo -Funktion	B
/F130/	Rückgängig gemachte Schritte wiederherstellen durch redo -Funktion	B
/F140/	Ausführen des nächsten Lösungsschrittes durch Hilfe-Funktion	B
/F150/	Berechnen des Lösungsweges ausgehend vom momentanen Zustand	A
/F160/	Automatisches Lösen eines Würfels bis zum Spielende	A
/F170/	Anzeigen des Spielmenüs	B
/F180/	Kontinuierliches Erstellen und automatisches Lösen eines Würfels (Bildschirmschonerfunktionalität)	A
/F190/	Programm beenden	B

5.2 Erweiterte Funktionen

Diese Funktionen sind notwendig, um auch die Wishkriterien zu realisieren.

Nr.	Beschreibung	Automatisch/ Benutzer
/F200/	Laden eines gespeicherten Spiels	B
/F210/	Ändern des Schwierigkeitsgrades	B
/F220/	Auswählen des Farbwürfels	B
/F230/	Auswählen des Themewürfels	B
/F240/	Ändern der Ton-Einstellungen (ein/aus)	B
/F250/	Ändern der Lautstärke	B
/F260/	Ändern der Sprach-Einstellungen	B
/F270/	Anzeigen der Highscore -Tabelle	A/B
/F280/	Speichern des aktuellen Spielzustandes	A/B
/F290/	Pause starten	A/B
/F300/	Pause beenden	B
/F310/	Berechnen der zum Lösen benötigten Zeit	A
/F320/	Berechnen der Punktezahl	A
/F330/	Prüfen, ob Punktezahl einen Highscore -Tabellenplatz bekommt	A
/F340/	Abspielen einer Melodie (wenn das Spiel gelöst oder ein Highscore-Tabellenplatz erreicht wurde)	A
/F350/	Einfügen der Punktezahl und des Namens in die Highscore -Tabelle	A

6 Produkt-Daten

6.1 System-Daten

- /D10/ Spielanleitung.
- /D20/ Tastenbelegung.
- /D30/ [Textur](#) - Dateien ([png](#)) für Würfel.
- /D40/ Sound- Dateien ([mid](#)) zur Wiedergabe.
- /D50/ Sprach- Dateien ([xml](#)) mit möglichen Übersetzungen in andere Sprachen .
- /D60/ Einstellungs-Datei ([xml](#)) zur Speicherung der Einstellungen.

6.2 Benutzer-Daten

- /D70/ [Highscore](#) (Wird nur benötigt, wenn das entsprechende Wunschkriterium erfüllt werden soll).
- /D80/ Daten zur Speicherung eines Spielstandes (Wird nur benötigt, wenn das entsprechende Wunschkriterium erfüllt werden soll).

7 Systemmodell

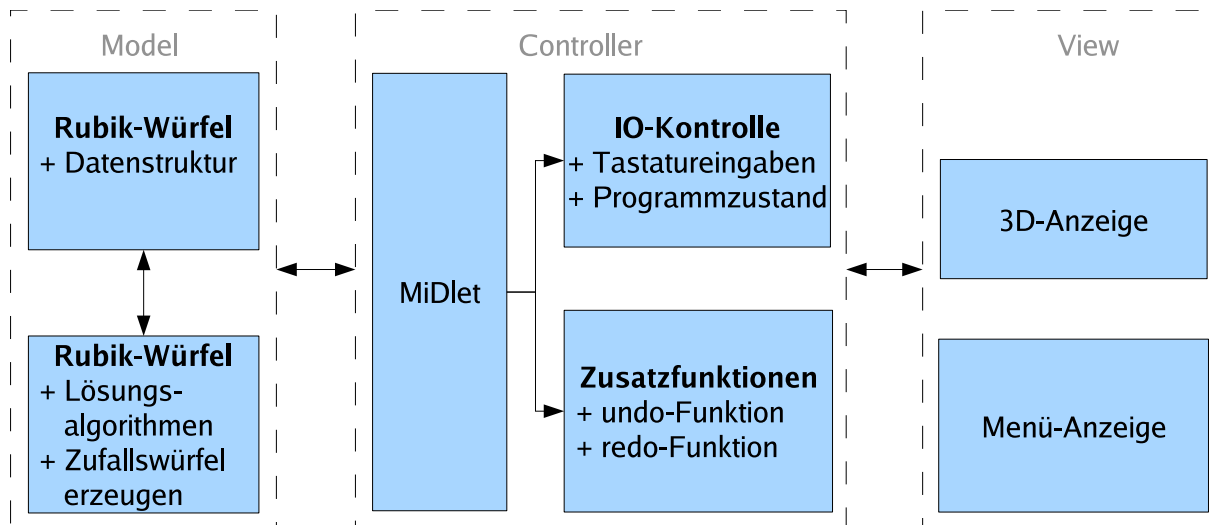


Abbildung 2 - Systemmodell

Die Grundstruktur dieses Programms basiert auf der [MVC](#)-Architektur. Dabei soll die Dreiteilung in Modell, View und Controller sowohl eine erleichterte Modifikation am Programmcode, als auch die Portierung auf andere Plattformen ermöglichen. Die drei Komponenten sollen dabei folgende Aufgaben erfüllen:

Model

Das Model enthält sowohl die Datenstruktur des Rubikwürfels, als auch die Algorithmen, die auf den Würfel angewandt werden.

Die Datenstruktur speichert die Ausgangsposition der Teilwürfel (aus denen der Rubikwürfel aufgebaut ist) und den aktuellen Zustand des Rubikwürfels.

An Algorithmen werden zum einen die Lösungsalgorithmen zum kompletten und schrittweisen Lösen des Würfels benötigt, zum anderen eine Funktion für das Erstellen eines zufällig verdrehten Würfels.

Controller

Der Controller ist die Zentraleinheit des Systems. Er koordiniert den Start, das Unterbrechen und das Beenden des Handyspiels. Ebenfalls koordiniert er die Tastatureingaben, die abhängig vom jeweiligen Programmzustand eine entsprechende Reaktion im Model bzw. View bewirken. Zuletzt übernimmt der Controller auch die Zusatzfunktionen wie z.B. die [undo](#) - und [redo](#) - Funktionen, oder eventuelle Wunschfunktionen (z.B. Tonwiedergabe).

View

Die View stellt dem Programm die Ausgabe auf dem Bildschirm ([LCD](#) des Handys) dar. Dabei gibt es die zweidimensionale Darstellung der Menüs (wie z.B. Hauptmenü siehe [9.1](#)) und die dreidimensionale Anzeige des Würfels im Spielmodus (siehe [9.3](#)).

8 Produkt-Leistungen

- /L10/ Das Drehen einer Scheibe muss für den Benutzer sichtbar sein und es muss innerhalb weniger Sekunden flüssig vollzogen werden.
- /L20/ Der Benutzer muss zum Drehen der Ebenen höchstens zwei Tasten betätigen.
- /L30/ Nach Aufruf der Hilfefunktion muss sich die entsprechende Scheibe sofort wie in /L10/ beschrieben drehen.
- /L40/ Durch Betätigen der **undo** -Taste können mindestens die letzten drei Zustände des Würfels wieder hergestellt werden.
- /L50/ Durch Betätigen der **redo** -Taste können bis zu drei zuvor rückgängig gemachten Schritte wieder hergestellt werden.
- /L60/ Das Erstellen eines zu lösenden Würfels erfolgt innerhalb von drei Sekunden.
- /L70/ Das Lösen eines Würfels erfolgt mit einem effizienten Algorithmus.
- /L80/ Der Speicherplatz umfasst höchstens 1 MB.

9 Benutzeroberfläche

9.1 Einführung

Die Benutzeroberfläche wird so aufgebaut sein, dass auch ein Benutzer ohne Vorkenntnisse den Sinn des Spiels erfassen kann. Der Benutzer kann sich vor Spielbeginn mit der Spielanleitung und der Tastenbelegung vertraut machen. Diese Informationen findet der Benutzer im wohlstrukturierten Menü.

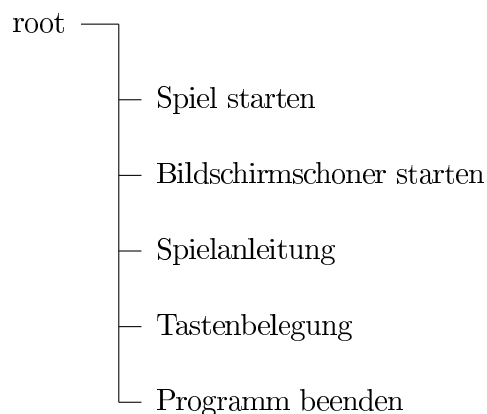
Das Spiel wird so gestaltet sein, dass es nur mit den Handytasten gelöst werden kann. Dabei wird die Tastenbelegung so gewählt, dass sie für den Benutzer nachvollziehbar ist und er intuitiv die richtige Belegung wählt. Für den Benutzer wird es also möglich sein, den Würfel ohne großen Aufwand optimal zu bewegen.

Alle Angaben zur die Benutzeroberfläche sind vorläufig, da auch die Wunschkriterien [2.2](#) enthalten sind.

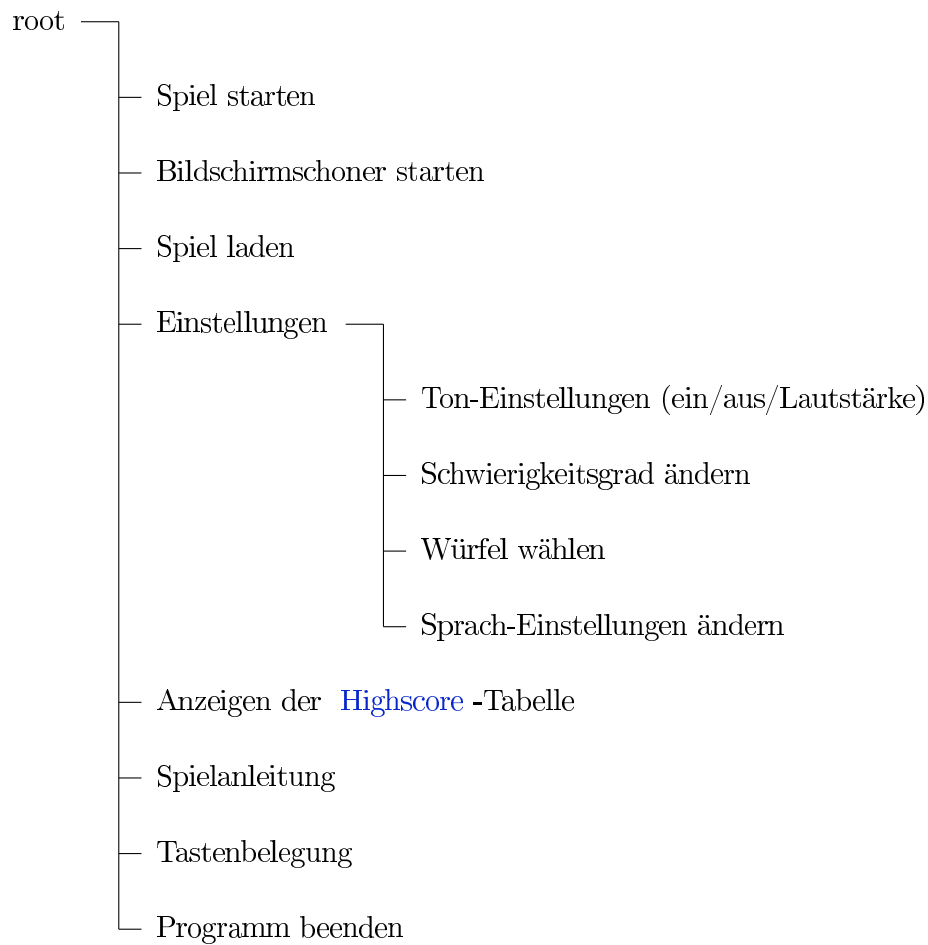
9.2 Startmenü

Nach Programmstart erscheint zuerst der Startbildschirm (Abbildung 3). Nach 2-3 Sekunden erscheint dann das Startmenü (Abbildung 4). Dieses bietet die Auswahlmöglichkeiten (Baumstruktur):

9.2.1 Nur Grundfunktionen



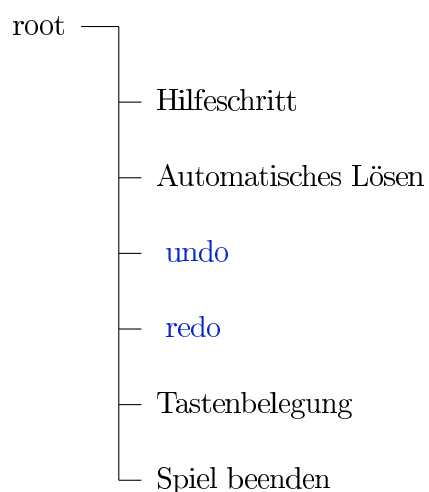
9.2.2 Mit erweiterten Funktionen



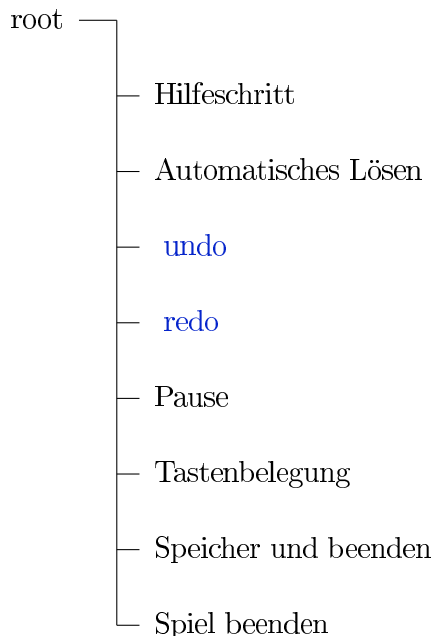
9.3 Spielmenü

Das Spielmenü (Abbildung 6) unterscheidet sich vom Startmenü dadurch, dass es vom Benutzer aus dem Spiel heraus aufgerufen wird, anders angezeigt (kein Vollbildmodus) wird und auch andere Menüpunkte enthält.

9.3.1 Nur Grundfunktionen



9.3.2 Mit erweiterten Funktionen



Die Benutzeroberfläche im Spielmodus stellt eine besondere Herausforderung dar, da sich der "Benutzer" keine Gedanken machen sollte, wie der Würfel und die Ebenen zu drehen sind.

Damit der Benutzer einen geeigneten Überblick über den Würfel hat, wird dieser als ein 3D-Objekt realisiert, das man mit Hilfe der vier Richtungstasten bewegen kann (Abbildung 5).

Da sich auf jeder der drei Raumachsen von Rubiks Würfel jeweils drei Ebenen befinden, werden insgesamt neun Tasten benötigt, um die gewünschte Ebene auf der korrekten Achse zu wählen. Es werden dafür die Ziffern (1-9) des Handys verwendet. Damit man weiß, mit welcher Taste man welche Ebene dreht, werden nummerierte Pfeile am Würfel dargestellt.

Anschließend gibt es jeweils noch zwei Richtungen, in die man die Ebene der jeweiligen Würfelachse verdrehen kann. Diese Richtung wird mittels der Stern- bzw. Rautetaste ausgewählt. Dabei bedeutet das Drücken der Sterntaste: Drehen in Pfeilrichtung. Durch Drücken der Rautetaste wird gegen die Pfeilrichtung gedreht. Die Drehung wird gleich ausgeführt.

Ein Spielzug besteht also aus dem Drücken von zwei Tasten:

1. Ebene und Achse auswählen (Nummerntaste 1-9)
2. Richtung wählen (Stern- bzw. Rautetaste)

Siehe dazu Abbildung 9.

Anmerkung:

Es ist zu beachten, dass die beschriebene Benutzeroberfläche, wie auch die folgenden Screenshots nur eine erste Näherung darstellen.

Die folgenden Screenshots entsprechen nicht der endgültigen Menüführung. Bei Abweichungen ist die Menüführung unter 9.2 und 9.3 maßgeblich.

9.4 Screenshots



Abbildung 3: Startbildschirm



Abbildung 4: Startmenü



Abbildung 5: Einstellungen



Abbildung 6: Spielmenü



Abbildung 7: Texturenwürfel

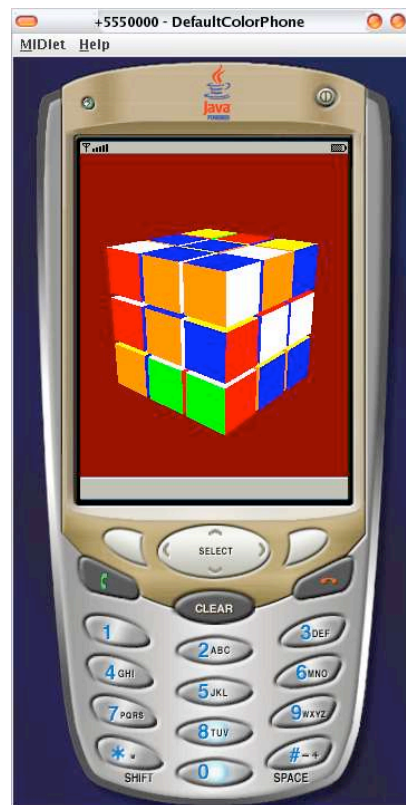


Abbildung 8: Bildschirmschoner



Abbildung 9: Spielmodus

10 Qualitätszielbestimmungen

- Benutzerfreundlichkeit wird durch eine möglichst einfache und intuitive Handhabung gewährleistet.
- Die Korrektheit sowohl der Hilfefunktion, als auch der ggf. vorhandenen Übersetzungen wird sicher gestellt.
- Die Effizienz steht im Hinblick auf geringe Wartezeiten auch im Vordergrund, denn das Spiel soll nicht träge wirken.
- Auch im schlimmsten Fall werden andere Komponenten (wie z.B. das Telefonbuch) nicht beschädigt. Diese Zuverlässigkeit wird durch das [Sandboxprinzip](#) der [JRE](#) gewährleistet.
- Die Portierbarkeit ist durch die verwendete Java API gegeben.
- Robustheit spielt bei der Entwicklung eine maßgebliche Rolle, so dass willkürliches Tastendrücken nicht zum Absturz führt.
- Wart-und Wiederverwendbarkeit ergibt sich durch die Implementierung im [MVC](#) -Prinzip.

11 Testfälle und Testszenarien

11.1 Testfälle

Für jede Funktion gibt es (mind.) einen Testfall. Ein Testfall ist ein aus Benutzersicht atomarer Vorgang. Da zwischen Grund- und erweiterten Funktionen unterschieden wird, wird auch hier zwischen Basis-Testfällen und erweiterten Testfällen unterschieden.

Basis-Testfälle

- /T010/ Das Programm starten. (/F010/, /F020/ [5.1](#))
- /T020/ Aufrufen der Spielanleitung. (/F030/ [5.1](#))
- /T030/ Aufrufen der Tastenbelegung aus dem Startmenü. (/F040/ [5.1](#))
- /T040/ Ein neues Spiel beginnen. (/F050/, /F070/, /F080/ [5.1](#))
- /T050/ Freies Drehen des Würfels. (/F090/ [5.1](#))
- /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung. (/F100/ [5.1](#))
- /T070/ Drehen der ausgewählten Ebene. (/F110/ [5.1](#))
- /T080/ Aufrufen des Spielmenüs. (/F170/ [5.1](#))
- /T090/ Aufrufen der Tastenbelegung aus dem Spielmenü. (/F040/ [5.1](#))
- /T100/ Aufrufen der Hilfefunktion. (/F150/, /F140/ [5.1](#))
- /T110/ Aufrufen des automatischen Lösens. (/F150/, /F160/ [5.1](#))
- /T120/ Rückgängig machen eines ausgeführten Zuges. (/F120/ [5.1](#))
- /T130/ Zuvor rückgängig gemachten Zug wiederherstellen. (/F130/ [5.1](#))
- /T140/ Spiel beenden. (/F060/ [5.1](#))
- /T150/ Aufrufen der Bildschirmschonerfunktionalität. (/F180/ [5.1](#))
- /T160/ Beenden des Programmes. (/F190/ [5.1](#))

Erweiterte Testfälle

- /T170/ Laden eines gespeicherten Spiels. (/F200/ [5.2](#))
- /T180/ Ändern der Ton-Einstellungen (ein/aus). (/F240/ [5.2](#))
- /T190/ Ändern des Schwierigkeitsgrades. (/F210/ [5.2](#))
- /T200/ Auswählen des [Farbwürfels](#) . (/F220/ [5.2](#))
- /T210/ Auswählen des [Themewürfels](#) . (/F230/ [5.2](#))
- /T220/ Sprach-Einstellungen ändern. (/F260/ [5.2](#))
- /T230/ Ändern der Lautstärke. (/F250/ [5.2](#))
- /T240/ Anzeigen der [Highscore](#) -Tabelle. (/F270/ [5.2](#))
- /T250/ Spiel pausieren. (/F290/ [5.2](#))

- /T260/ Pause beenden. (/F300/ 5.2)
- /T270/ Speichern des aktuellen Spielzustandes und Beenden des Spiels. (/F280/, /F060/ 5.2)
- /T280/ Erreichen eines Highscore-Tabellenplatzes. (/F310/, /F320/, /F330/, /F340/ 5.2)
- /T290/ Eintragen in die Highscore-Tabelle. (/F350/ 5.2)

Stabilitätstests

- /T300/ Anruf während Ausführung des Programms.
- /T310/ Empfang einer SMS während Ausführung des Programms.
- /T320/ Stresstest durch "wildes" Drücken von Tasten.

11.2 Testszenarien

Die Testszenarien setzen sich aus den genannten Testfällen zusammen. Mit Stern (*) markierte Szenarien enthalten erweiterte Testfälle bzw. Funktionen.

Testszenario 1 - Ein normales Spiel

Ein unerfahrener Benutzer startet das Spiel und möchte (unter Zuhilfenahme aller verfügbaren Spielhilfen) einen Würfel lösen, was er schließlich auch schafft.

1. /T010/ Das Programm starten.
2. /T020/ Aufrufen der Spielanleitung.
3. /T030/ Aufrufen der Tastenbelegung aus dem Startmenü.
4. /T040/ Ein neues Spiel beginnen.
5. /T050/ Freies Drehen des Würfels.
6. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
7. /T070/ Drehen der ausgewählten Ebene.
8. /T080/ Aufrufen des Spielmenüs.
9. /T100/ Aufrufen der Hilfefunktion.
10. /T090/ Aufrufen der Tastenbelegung aus dem Spielmenü.
11. /T120/ Rückgängig machen eines ausgeführten Zuges.
12. /T130/ Zuvor rückgängig gemachten Zug wiederherstellen.
13. Wiederholen von 5. bis 9. bis zur Lösung des Würfels.
14. /T160/ Beenden des Programmes.

Testszenario 2 - Vollautomatisches Lösen

Der Benutzer startet das Spiel, lässt den Würfel aber nach ein paar Versuchen automatisch lösen.

1. /T010/ Das Programm starten.
2. /T040/ Ein neues Spiel beginnen.

3. /T050/ Freies Drehen des Würfels.
4. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
5. /T070/ Drehen der ausgewählten Ebene.
6. Wiederholen von 3. bis 5.
7. /T080/ Aufrufen des Spielmenüs.
8. /T110/ Aufrufen des automatischen LöSENS.
9. /T160/ Beenden des Programmes.

Testszenario 3 - Dauerhilfe

Der Benutzer startet ein neues Spiel und löst dieses ausschließlich durch wiederholtes Aufrufen der Hilfefunktion.

1. /T010/ Das Programm starten.
2. /T040/ Ein neues Spiel beginnen.
3. /T080/ Aufrufen des Spielmenüs.
4. /T100/ Aufrufen der Hilfefunktion.
5. Wiederholen von 3. und 4. bis zur Lösung des Würfels.
6. /T160/ Beenden des Programmes.

Testszenario 4 - Alles rückgängig

Der Benutzer beginnt ein neues Spiel und macht genau zwei Züge. Dann möchte er durch wiederholtes Aufrufen der **undo** -Funktion alle seine Züge rückgängig machen, wobei er versucht, die **undo** -Funktion öfter als zwei mal aufzurufen. Im Anschluss daran ruft er das automatische LöSEN auf, welches er aber, nachdem er es gestartet hat, abubrechen versucht.

1. /T010/ Das Programm starten.
2. /T040/ Ein neues Spiel beginnen.
3. /T050/ Freies Drehen des Würfels.
4. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
5. /T070/ Drehen der ausgewählten Ebene.
6. Einmaliges Wiederholen von 3. bis 5.
7. /T080/ Aufrufen des Spielmenüs.
8. /T120/ Rückgängig machen eines ausgeführten Zuges.
9. Mind. drei mal Wiederholen von 7. und 8.
10. /T080/ Aufrufen des Spielmenüs.
11. /T110/ Aufrufen des automatischen LöSENS.
12. Drücken aller Tasten, die aus Benutzersicht evtl. zum (nicht möglichen) "Abbrechen" der Aktion führen könnten.

13. /T160/ Beenden des Programmes.

Testszenario 5 - Bildschirmschoner

Der Benutzer beginnt ein Spiel, welches er aber abbricht, um danach den Bildschirmschoner zu starten. Nachdem der Bildschirmschoner mind. 10 Minuten gelaufen ist, beendet er diesen und danach das Programm.

1. /T010/ Das Programm starten.
2. /T040/ Ein neues Spiel beginnen.
3. /T050/ Freies Drehen des Würfels.
4. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
5. /T070/ Drehen der ausgewählten Ebene.
6. Wiederholen von 3. bis 5.
7. /T080/ Aufrufen des Spielmenüs.
8. /T140/ Spiel beenden.
9. /T150/ Aufrufen der Bildschirmschonerfunktionalität.
10. Mind. 10 Minuten warten
11. /T160/ Beenden des Programmes.

Testszenario 6 - Highscore geknackt*

Der Benutzer betrachtet nach Programmstart die (nicht leere) [Highscore](#)-Tabelle, stellt den höchsten Schwierigkeitsgrad ein und beginnt dann ein neues Spiel. Nach ein paar Zügen pausiert er das Spiel für drei Minuten und setzt es danach fort. Er löst den Würfel in "Rekordzeit", eine Melodie ertönt und er erhält den höchsten Rang in der Highscore-Tabelle. Zufrieden beendet er das Programm.

1. /T010/ Das Programm starten.
2. /T240/ Anzeigen der Highscore-Tabelle.
3. /T190/ Ändern des Schwierigkeitsgrades.
4. /T040/ Ein neues Spiel beginnen.
5. /T050/ Freies Drehen des Würfels.
6. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
7. /T070/ Drehen der ausgewählten Ebene.
8. Wiederholen von 5. bis 7.
9. /T250/ Spiel pausieren.
10. Drei Minuten warten.
11. /T260/ Pause beenden.
12. Wiederholen von 5. bis 7.
13. /T280/ Erreichen eines Highscore-Tabellenplatzes.

14. /T290/ Eintragen in die Highscore-Tabelle.

15. /T160/ Beenden des Programmes.

Testszenario 7 - Speichern und Laden*

Der Benutzer beginnt ein neues Spiel mit zuvor selbst gewählten Themenwürfel. Nach ein paar Zügen und der Benutzung der Hilfefunktion speichert er den Spielstand und beendet sowohl Spiel als auch Programm. Nach erneutem Programmstart lädt er das gespeicherte Spiel und setzt es fort.

1. /T010/ Das Programm starten.
2. /T210/ Auswählen des [Themewürfels](#) .
3. /T040/ Ein neues Spiel beginnen.
4. /T050/ Freies Drehen des Würfels.
5. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
6. /T070/ Drehen der ausgewählten Ebene.
7. Wiederholen von 4. bis 6.
8. /T080/ Aufrufen des Spielmenüs.
9. /T100/ Aufrufen der Hilfefunktion.
10. /T270/ Speichern des aktuellen Spielzustandes und Beenden des Spiels.
11. /T160/ Beenden des Programmes.
12. /T010/ Das Programm starten.
13. /T170/ Laden eines gespeicherten Spiels.
14. /T050/ Freies Drehen des Würfels.
15. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
16. /T070/ Drehen der ausgewählten Ebene.
17. Wiederholen von 14. bis 16. bis zum Lösen des Würfels.
18. /T160/ Beenden des Programmes.

Testszenario 8 - Wunschfarben und -lautstärke*

Der Benutzer wählt vor Spielstart den Farbwürfel aus, setzt die Lautstärke auf ein Maximum und schaltet - paradoxerweise - den Ton dann aus. Beim Lösen des folgenden normalen Spiels ist dann auch keine Melodie zu hören.

1. /T010/ Das Programm starten.
2. /T200/ Wählen des Farbwürfels.
3. /T230/ Ändern der Lautstärke.
4. /T180/ Ändern der Ton-Einstellungen (aus).
5. /T040/ Ein neues Spiel beginnen.

6. /T050/ Freies Drehen des Würfels.
7. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
8. /T070/ Drehen der ausgewählten Ebene.
9. Wiederholen von 6. bis 8. bis zum Lösen des Würfels.
10. /T160/ Beenden des Programmes.

Testszenario 9 - Anruf*

Der Benutzer startet ein normales Spiel. Während des Spiels wird er angerufen. Nach dem Anruf möchte der Benutzer das durch den Anruf automatisch pausierte Spiel fortsetzen.

1. /T010/ Das Programm starten.
2. /T040/ Ein neues Spiel beginnen.
3. /T050/ Freies Drehen des Würfels.
4. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
5. /T070/ Drehen der ausgewählten Ebene.
6. Wiederholen von 3. bis 5.
7. /T310/ Anruf während Ausführung des Programms.
8. Telefonat beenden.
9. Pause beenden.
10. /T050/ Freies Drehen des Würfels.
11. /T060/ Auswählen einer zu drehenden Ebene und der Drehrichtung.
12. /T070/ Drehen der ausgewählten Ebene.
13. Wiederholen von 12. bis 14. bis zum Lösen des Würfels.
14. /T160/ Beenden des Programmes.

Testszenario 10 - Mehrsprachig*

Der Benutzer möchte die Standardsprache (Deutsch) des Spieles auf Chinesisch ändern. Ist kein Chinesisch vorhanden, so wählt der Benutzer Englisch aus und spielt ein normales Spiel (Testszenario 1). Hierbei werden jetzt alle sprachlichen Elemente (Menüs, Anleitungen etc.) auf Englisch angezeigt.

1. /T010/ Das Programm starten.
2. /T220/ Sprach-Einstellungen ändern.
3. Weiter bei Testszenario 1 ab Schritt 2.

12 Entwicklungs-Umgebung

12.1 Software

- Betriebssysteme
 - Windows 2000/XP
 - Fedora Core 4/5/6
- Modellierung
 - Visual Paradigm for UML
- Entwicklung
 - [Eclipse](#) 3.1.x / 3.2.x
 - Die [J2ME](#) -Laufzeitumgebung mit [CLDC](#) 1.1 Konfiguration und [MIDP](#) 2.0 Profil.
 - Die Mobile3D Graphics API (JSR 184 Mobile 3D Graphics) für [J2ME](#) .
 - [KToolbar](#)
- Versionsverwaltung
 - [Subversion](#)
- Sonstige Software
 - \LaTeX (für Dokumentation)
 - GIMP (Grafiken/Screenshots)
 - Wiki-System (knowledge management)

12.2 Hardware

- Diverse Standard-PCs (mit Intel/AMD CPUs)
- Diverse [CLDC](#) 1.1 kompatible Handys

Glossar

Bestenliste	Siehe Highscore - Liste, 4
CLDC	(engl. Connected Limited Device Configuration) definiert die kleinstmögliche Konfiguration einer J2ME-Laufzeitumgebung., 7 , 24
Eclipse	Eclipse ist eine Entwicklungsumgebung (IDE) für die Programmiersprache Java., 24
Farbwürfels	Der Standardwürfel mit den sechs Standardfarben weiß, blau, rot, grün, gelb und orange., 8 , 18
flüchtiger Speicher	Speicher, dessen Inhalte verlorengehen, wenn er nicht mit Strom versorgt wird, also v.a. das RAM (Arbeitsspeicher)., 7
GUI	(engl. Graphical User Interface, dt. graphische Benutzerschnittstelle) Eine grafische Benutzeroberfläche ist eine Softwarekomponente, die einem Computerbenutzer die Interaktion mit der Maschine über grafische, metaphorhafte Elemente (Arbeitsplatz, Symbole, Papierkorb, Menü) unter Verwendung eines Zeigeegerätes (wie einer Maus) erlaubt., 4
Highscore	Eine Highscore-Tabelle bezeichnet in Computerspielen bzw. Handyspielen eine Tabelle, in der die besten Spieler nach ihren in einer Spielpartie erreichten Punkten (Highscores) absteigend sortiert dargestellt werden., 4 , 8 , 9 , 12 , 18 , 21 , 25
J2ME	Java 2 Platform, Micro Edition, abgekürzt J2ME, ist eine Umsetzung der Programmiersprache Java für so genannte "embedded consumer products" wie etwa Mobiltelefone oder PDAs. Definiert wird es in den JSR 30 und 37., 7
Java Virtual Machine (JVM)	Die Java Virtual Machine (abgekürzt Java VM oder JVM) ist der Teil der Java-Laufzeitumgebung (JRE) für Java-Programme, der für die Ausführung des Java-Bytecodes verantwortlich ist. Der andere Teil der Java-Laufzeitumgebung sind die Java-Klassenbibliotheken. Die JVM dient dabei als Schnittstelle zur Maschine und zum Betriebssystem und ist für die meisten Plattformen verfügbar (z. B. Linux, Mac, Palm OS, Solaris, Windows, usw.)., 7
JRE	JRE steht im engeren Sinne für Java Runtime Environment (deutsch Java-Laufzeitumgebung) des US-Unternehmens Sun Microsystems. Sie liefert unter anderem die Java VM. Sie wird benötigt, um Java-Applikationen auszuführen., 17

KToolbar	KToolbar ist eine Testumgebung für die Entwicklung von J2ME fähigen Handy-Anwendungen. Dabei simuliert KToolbar ein Handy, 4 , 7 , 24
Lösungsalgorithmus	Unter einem Lösungsalgorithmus versteht man allgemein eine genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen., 4
LCD	Ein Flüssigkristallbildschirm (englisch liquid crystal display, kurz LCD), ist ein Bildschirm, bei dem spezielle Flüssigkristalle zur Bilddarstellung genutzt werden, die die Polarisationsrichtung von Licht beeinflussen können. Diese Bildschirme, deren Realisierung nur mit einer Matrix von Dünnschichttransistoren (TFT: thin-film transistor) möglich ist, stellen die zurzeit dominante Flachbildschirm-Technologie dar., 10
mid	Ein Dateiformat für MIDI-Daten (*.mid): MIDI (engl.: musical instrument digital interface = "Digitale Schnittstelle für Musikinstrumente") ist ein Datenübertragungs-Protokoll zum Zwecke der Übermittlung, Aufzeichnung und Wiedergabe von musikalischen Steuerinformationen zwischen digitalen Instrumenten oder mit einem PC., 9
MIDP	(Mobile Information Device Profile) ist ein Profil der Java 2, Micro Edition (J2ME), das speziell auf die Fähigkeiten kleiner mobiler Endgeräte wie Mobiltelefon oder Pager ausgelegt ist. Es umfasst daher Funktionen zur Ansteuerung und Abfrage von ITU-T Einhandtastaturen - Miniaturbildschirmen - flüchtigen und nicht-flüchtigen Speicher im Kilobyte-Bereich etc. ., 7
Model-View-Controller-Prinzip	abgekürzt: MVC. Der Begriff Model-View-Control (MVC) bezeichnet ein Architekturmuster zur Aufteilung von Softwaresystemen in die drei Einheiten Datenmodell (engl. Model), Präsentation (engl. View) und Programmsteuerung (engl. Controller)., 25
MVC	siehe Model-View-Controller-Prinzip , 10 , 17
Piktogrammen	Ein Piktogramm (von lat.: pictum = gemalt, Bild; griech.: graphein = schreiben) ist ein einzelnes Bildsymbol, das eine Information durch vereinfachte grafische Darstellung vermittelt., 6

png	PNG steht für Portable Network Graphics und ist ein Grafikformat für Rastergrafiken. Es wurde als freier Ersatz für das ältere proprietäre Format GIF entworfen und ist weniger komplex als TIFF. Die Daten werden verlustfrei komprimiert abgespeichert (im Gegensatz zum verlustbehafteten JPEG-Dateiformat)., 9
QQVGA	Steht für "Quater Quater VGA " mit einer Größe von 160 x 120 Pixel, 7
redo	Bezeichnet die Möglichkeit, nach einem Rückgängigmachen (siehe undo) dieselben Schritte ohne Neueingabe nochmals ausführen zu lassen., 4 , 8 , 10 , 11 , 13
Sandboxprinzip	Sandbox ist der englischsprachige Begriff für Sandkiste oder Sandkasten. Im Rahmen der Informationstechnik steht er für Besonderheiten der Laufzeitumgebung einer Software. Die Software wird vom Rest des Systems abgeschirmt, quasi in den Sandkasten gesetzt, in dem es einerseits keinen Schaden anrichten kann und andererseits die Wirkungen der Software aufgezeichnet werden können., 17
Subversion	Subversion (SVN) ist eine Open-Source-Software zur Versionsverwaltung. Es wurde als moderne Ablösung für das Programm CVS entwickelt., 24
Textur	Bei der 3D-Computer-Animation, bzw. Visualisierung, Effekte, usw. bezeichnet Textur das Bild, welches auf der Oberfläche eines virtuellen Körpers dargestellt wird; man spricht von Texture Mapping., 4 , 6 , 9
Themewürfels	Statt Farben sind auf dem sog. Themewürfel sind - im Gegensatz zum Farbwürfel - einfache und kontrastreiche Texturen statt der Farben vorhanden. So wird ermöglicht, dass das Spiel auch auf einem Handy ohne Farbdisplay oder von Personen mit Rot-Grün-Sehschwäche gespielt werden kann., 8 , 18 , 22
Touchscreen	Ein Touchscreen, Tastschirm bzw. Sensorbildschirm ist ein Computerbildschirm mit Bereichen, durch deren Berührung der Programmablauf gesteuert werden kann. Statt einen Cursor per Maus o.ä. zu steuern, kann der Finger oder ein Zeigestift verwendet werden., 7

undo	(englisch für rückgängig machen) Ist die Bezeichnung für die Funktion von Computerprogrammen, eine oder mehrere Eingaben zurückzunehmen. Je nach Komplexität der Anwendung findet sich keine Undo-Möglichkeit oder ein nur einstufiges Undo bis hin zur Rücknahme sämtlicher Arbeitsschritte., 4 , 8 , 10 , 11 , 13 , 20 , 25
VGA	Der Begriff Video Graphics Array (VGA) bezeichnet einen Computergrafik-Standard (EISA, 1987), der bestimmte Kombinationen von Bildauflösung und Farbanzahl (Bittiefe) sowie Wiederholfrequenz definiert und praktisch identisch zum MCGA (Multicolor Graphics Adapter) von IBM ist., 25
xml	Die Extensible Markup Language (dt. erweiterbare Auszeichnungssprache), abgekürzt XML, definiert Regeln für den Aufbau von Dokumenten, die Daten enthalten, die zum Teil einer fest vorgegebenen Struktur entsprechen, teilweise aber auch Elemente beinhalten, die nicht diesem statischen Schema entsprechen., 9