

Inhaltsverzeichnis

1. Einleitung	3
2. Tests	4
2.1. Übersicht	4
2.2. Werkzeuge	4
2.2.1. Manuelle Unterstützung	4
2.2.2. Automatisiertes Testen	5
2.3. Pflichtenheft-Verweise	5
2.4. Protokolle	6
2.4.1. Funktionstests	6
2.4.2. Komponententests	7
2.4.3. Negativtests	8
2.4.4. Extremtests	8
2.4.5. Abnahmetests	8
2.4.6. Nicht durchgeführte Tests	8
2.5. Statistik	8
2.5.1. Testabdeckung durch Komponententests	8
3. Programmfehler	9
3.1. Übersicht	9
3.1.1. Klassifizierung	9
3.2. Werkzeuge	9
3.2.1. Manuelle Unterstützung	9
3.2.2. Automatisierte Prüfung	9
3.3. Statistik	9
4. Änderungen	10
4.1. Protokoll	10
4.1.1. Behobene Probleme	10
4.1.2. Nicht behobene Probleme	10
4.1.3. Verschönerungen	10
5. Ausnahmen	11
5.1. Behandlungen	11
5.2. Meldungen	11
6. Abschluss	12
6.1. Bewertung	12

A. Anhang	13
A.1. Vollständige Fehlerliste	13
A.2. Programmaufnahmen	13

1. Einleitung

2. Tests

2.1. Übersicht

2.2. Werkzeuge

Zur Testdurchführung nutzen wir einige Werkzeuge. Wichtig bei deren Wahl waren uns diese Kriterien:

- Für Studenten kostenlos, da es hier um ein studentisches Projekt geht.
- Aktuelle Open-Source-Software.
- Lokale Integrierbarkeit in unsere Entwicklungsumgebung.
- Verwendbarkeit zusammen mit dem Versions-Management-System Git über GitHub / Online ...

2.2.1. Manuelle Unterstützung

Lokal unter Visual Studio installierte Werkzeuge.

NUnit , *V. 2.6.3*

NUnit ist ein Framework für Komponententests für alle .NET-Sprachen.

Internetseite: <http://www.nunit.org/>

OpenCover , *V. 4.5.1923*

OpenCover ermittelt die Testabdeckung unter .NET-Sprachen ab Version 2.0.

Internetseite: <http://opencover.codeplex.com/>

ReportGenerator , V.

ReportGenerator erzeugt aus den von OpenCover produzierten XML-Daten einen übersichtlichen Bericht, z.B. in Form von HTML-Seiten.

Internetseite: <http://reportgenerator.codeplex.com/>

Jeweils sind NuGet Pakete verfügbar.

Für deren Zusammenspiel ist es nötig ein Skript zu schreiben. Unter Windows übernimmt diese Aufgabe bei uns eine einfache Stapelverarbeitungsdatei (Batch-Datei/.bat-Dateiendung). Alternativ und unter Voraussetzung weiterer Kenntnisse können spezielle Build-Skripte verwendet werden.

Einerseits war es uns wichtig, dass die Werkzeuge lokal bei jedem verfügbar und ausführbar sind. Andererseits haben wir z.B.

auch eine Automatisierung, da z.B. das individuelle Erstellen der Testabdeckungs-Statistiken sich ständig wiederholt und Zeit verschwendet.

2.2.2. Automatisiertes Testen

OpenCover

Eigener HTML-Generator

Travis CI

2.3. Pflichtenheft-Verweise

2.4. Protokolle

2.4.1. Funktionstests

FT_001 *Einstellung der Grafikauflösung.*

Die möglichen Einstellungen werden dynamisch vom Betriebssystem angefordert. D.h. die Werte, welche dem Spieler zur Auswahl stehen sind bereits vom Betriebssystem auf Gültigkeit überprüft worden (siehe: Microsoft.Xna.Framework.Graphics.SupportedDisplayModes).

FT_010 *Gültige Knoten-Transformationen.*

Wir definieren eine Liste möglicher Transformationen ausgehend vom Startknoten. Jede Transformation ist einzeln ausführbar.

1. Jede einzelne Kante des Startknotens ist selektierbar.
2. Mehrere Kanten (zwei, drei oder vier) des Startknotens sind selektierbar.
3. Jede einzelne Kante des Startknotens ist in jede Richtung des dreidimensionalen Raumes um einen Schritt durch direktes Anklicken und anschließendes Ziehen mit der Maus verschiebbar.
4. Jede einzelne Kante des Startknotens ist in jede Richtung des dreidimensionalen Raumes um mehrere (mindestens zehn) Schritte durch direktes Anklicken und anschließendes Ziehen mit der Maus verschiebbar.
5. Mehrere (mindestens zwei) selektierte Kanten sind um einen Schritt durch direktes Anklicken und anschließendes Ziehen mit der Maus verschiebbar.
6. Mehrere (mindestens zwei) selektierte Kanten sind um mehrere (mindestens zehn) Schritte durch direktes Anklicken und anschließendes Ziehen mit der Maus verschiebbar.
7. Jede einzelne Kante des Startknotens ist in jede Richtung des dreidimensionalen Raumes um einen Schritt durch Anklicken der Navigationspfeile und anschließendes Ziehen mit der Maus verschiebbar.
8. Jede einzelne Kante des Startknotens ist in jede Richtung des dreidimensionalen Raumes um mehrere (mindestens zehn) Schritte durch Anklicken der Navigationspfeile und anschließendes Ziehen mit der Maus verschiebbar.
9. Mehrere (mindestens zwei) selektierte Kanten sind um einen Schritt durch Anklicken der Navigationspfeile und anschließendes Ziehen mit der Maus verschiebbar.

10. Mehrere (mindestens zwei) selektierte Kanten sind um mehrere (mindestens zehn) Schritte durch Anklicken der Navigationspfeile und anschließendes Ziehen mit der Maus verschiebbar.
11. Der in ?? abgebildete, Knoten „Schlaufe“ ist erstellbar.
12. Der in ?? abgebildete, Knoten „Überleger“ ist erstellbar.
13. Jede einzelne Kante des Startknotens lässt sich nach ihrer Verschiebung in die vorige Position durch direktes Anklicken und anschließendes Ziehen zurücksetzen.
14. Jede einzelne Kante des Startknotens lässt sich nach ihrer Verschiebung in die vorige Position durch Anklicken des „Undo“-Buttons zurücksetzen.
15. Jede einzelne Kante des Startknotens lässt sich nach ihrer Verschiebung in die vorige Position durch Anklicken des „Undo“-Buttons zurücksetzen und der „Redo“-Button macht die Aktion des „Undo“-Buttons rückgängig.

FT_020 *Eine neue Challenge aus zwei im Creative-Mode erzeugten Knoten erstellen.*

1. Im Hauptmenü auf den Text „NEW Creative“ klicken.
2. Im folgenden Menü auf den Text „NEW Challenge“ klicken.
3. Im folgenden Menü in der linken Auswahlliste einen Zielknoten
4. In der rechten Liste einen Startknoten auswählen.
5. Im rechteckigen Eingabefeld einen Namen für die Challenge eingeben und mit bestätigen.

2.4.2. Komponententests

Wir führen für fast jede Komponente Tests durch. Davon ausgenommen sind:

Grafik-Komponenten

Daten

Zur Strukturierung der Test spiegeln wir das Projekt welches den Programmcode enthält. D.h. zu jeder Komponente die wir testen gibt es eine Testklasse im Tests-Projekt. Eine Statistik zur Testabdeckung durch Komponententests ist verfügbar (siehe 2.5.1).

2.4.3. Negativtests

2.4.4. Extremtests

2.4.5. Abnahmetests

2.4.6. Nicht durchgeführte Tests

2.5. Statistik

2.5.1. Testabdeckung durch Komponententests

3. Programmfehler

3.1. Übersicht

3.1.1. Klassifizierung

Bug Fehler im Programm.

Design Fehler in der Darstellung.

Missing Fehlender Bestandteil.

Question Frage zu einem möglichen Problem die zu diskutieren ist.

3.2. Werkzeuge

3.2.1. Manuelle Unterstützung

3.2.2. Automatisierte Prüfung

3.3. Statistik

4. Änderungen

4.1. Protokoll

4.1.1. Behobene Probleme

4.1.2. Nicht behobene Probleme

4.1.3. Verschönerungen

5. Ausnahmen

5.1. Behandlungen

5.2. Meldungen

6. Abschluss

6.1. Bewertung

A. Anhang

A.1. Vollständige Fehlerliste

A.2. Programmaufnahmen