# Code Review

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 19.04.2012 | 1.0 | Erste Version des Dokuments, erstes Code Review | LE |

## Übersicht

Ziel dieses Dokumentes ist die Vorbereitung und Dokumentation von Code Reviews. Zuerst werden die Code Review Kriterien festgelegt, danach werden die einzelnen Reviews dokumentiert mit jeweiligen Kommentaren.

## Kriterien

### Code Style Analyse

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Code Richtlinien von HSR Video Wall wurden eingehalten |  |  |
| Der Code wirkt durch seine Anordnung & Verschachtelung übersichtlich |  |  |
| Die HSR Video Wall Headers sind in allen nicht generierten Sourcen vorhanden |  |  |
| Die XML-Kommentar Kompilation wurde in den Projekten aktiviert |  |  |
| Alle Public / Protected Members sind ausreichend Dokumentiert |  |  |
| Die auskommentierten Programm-Stücke sind ausreichend erklärt |  |  |
| Die Projekte enthalten keine toten Programm-Klassen |  |  |
| Die fehlenden Programmstücke sind mittels TODO-Kommentar beschrieben |  |  |
| Der Code übersetzt ohne Compiler Warnings |  |  |
| Es gibt keine Bad Smells im Code |  |  |

### Exception Handling

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| ?Der Code enthält keine abgefangenen und ignorierten Ausnahmen |  |  |
| ?Fehler in asynchronen Prozessen werden mittels Event weitergeleiltet |  |  |
| ?Das Logging erfasst alle Fehler aus allen Funktionalitätsschichten |  |  |
| Das Before/After Pattern wird, wo möglich, mittels using() { } angewendet |  |  |
| IDisposable.Dispose() Methoden werden in jedem Fall aufgerufen |  |  |

### Flow Control

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Es existieren keine Schleifen ohne Abbruchkriterien |  |  |
| Es existieren keine nicht dokumentierte, tote Programmstücke (z.B. if(false) / while (false) / …) |  |  |
| Rekursive Calls haben immer eine Verankerung und Abbruchbedingung |  |  |

### Naming

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Namen der Klassen / Variabeln sind selbstbeschreibend |  |  |
| Verwirrende oder falsche Namen sind nicht vorhanden |  |  |
| Interface-Klassen beginnen immer I (z.B. IDisposable) |  |  |
| Klassen / Properties / Methoden werden mit *PascalCasing* geschrieben |  |  |
| Lokale Variablen / Argumente werden mit *camelCasing* geschrieben |  |  |
| Konstanten werden in *PascalCasing* geschrieben |  |  |
| Felder in Klassen werden mit *\_camelCasing* geschrieben |  |  |
| Der Code enthält keine Magic Numbers |  |  |

### Tools

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Warnings von Resharper 4.XXX werden, wo sinnvoll, behoben |  |  |
| Die Errors von FxCop werden wo sinnvoll behoben |  |  |

## Code Review 19.04.2012

Das erste Code Review wurde am 19.04.2012 durchgeführt. Anwesend waren Christina Heidt, Delia Treichler und Lukas Elmer. Es wurde zuerst im Speziellen eine Klasse untersucht, die für die Umrechnung der Maus Position zwischen der Kinect Hand Position und dem Fenster zuständig ist. Danach wurden die Kriterien Punkt für Punkt bewertet.

Das Review zieht einige Verbesserungen nach sich, die sich aus den negativ bewerteten Kriterien ergeben. Im Speziellen sind das:

* XML Dokumentation im Code, Headers
* Genaue Untersuchung, ob die IDisposable.Dispose() Methoden immer aufgerufen werden
* Es wurden „Magic Numbers“ gefunden, die dokumentiert und werden müssen ausgelagert (in statisches Attribut oder in Konfiguration).

### Code Style Analyse

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Code Richtlinien von HSR Video Wall wurden eingehalten | X |  |
| Der Code wirkt durch seine Anordnung & Verschachtelung übersichtlich | X |  |
| Die HSR Video Wall Headers sind in allen nicht generierten Sourcen vorhanden |  | X |
| Die XML-Kommentar Kompilation wurde in den Projekten aktiviert |  | X |
| Alle Public / Protected Members sind ausreichend Dokumentiert |  | X |
| Die auskommentierten Programm-Stücke sind ausreichend erklärt | X |  |
| Die Projekte enthalten keine toten Programm-Klassen | X |  |
| Die fehlenden Programmstücke sind mittels TODO-Kommentar beschrieben | X |  |
| Der Code übersetzt ohne Compiler Warnings | X |  |
| Es gibt keine Bad Smells im Code | X |  |

### Exception Handling

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| ?Der Code enthält keine abgefangenen und ignorierten Ausnahmen |  |  |
| ?Fehler in asynchronen Prozessen werden mittels Event weitergeleiltet |  |  |
| ?Das Logging erfasst alle Fehler aus allen Funktionalitätsschichten |  |  |
| Das Before/After Pattern wird, wo möglich, mittels using() { } angewendet | X |  |
| IDisposable.Dispose() Methoden werden in jedem Fall aufgerufen |  | ? |

### Flow Control

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Es existieren keine Schleifen ohne Abbruchkriterien | X |  |
| Es existieren keine nicht dokumentierte, tote Programmstücke (z.B. if(false) / while (false) / …) | X |  |
| Rekursive Calls haben immer eine Verankerung und Abbruchbedingung | X |  |

### Naming

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Namen der Klassen / Variabeln sind selbstbeschreibend | X |  |
| Verwirrende oder falsche Namen sind nicht vorhanden | X |  |
| Interface-Klassen beginnen immer I (z.B. IDisposable) | X |  |
| Klassen / Properties / Methoden werden mit *PascalCasing* geschrieben | X |  |
| Lokale Variablen / Argumente werden mit *camelCasing* geschrieben | X |  |
| Konstanten werden in *PascalCasing* geschrieben | X |  |
| Felder in Klassen werden mit *\_camelCasing* geschrieben | X |  |
| Der Code enthält keine Magic Numbers |  | X |

### Tools

|  |  |  |
| --- | --- | --- |
|  | Erfüllt | N. Erfüllt |
| Die Warnings von Resharper 4.XXX werden, wo sinnvoll, behoben | X |  |
| Die Errors von FxCop werden wo sinnvoll behoben | X |  |