

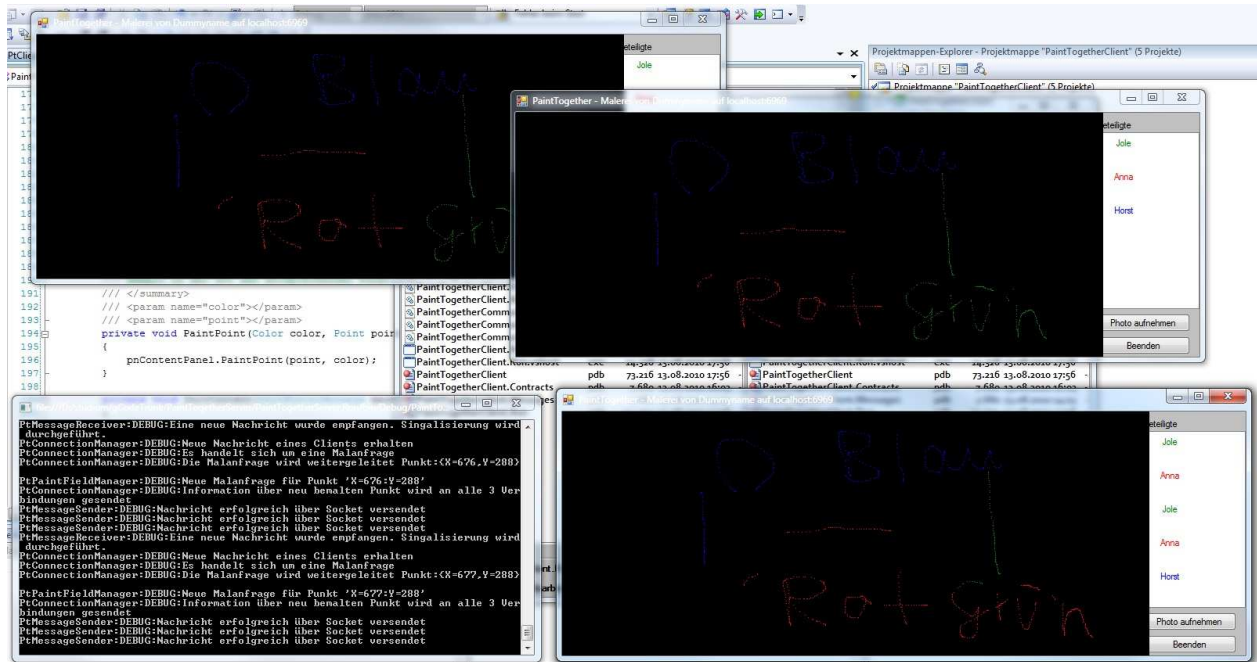
Fach: Verteilte Informationsverarbeitung
Studienheft: VII01-03
Projekt: Client-Server-Malprogramm
Dokumenttitel: Phase 4.1: Test
Dokumentart: Dokumentation
Erstellt von: Martin Blankenstein
Erstelldatum: 26.08.2010

Inhaltsverzeichnis:

| | |
|---|----|
| 1 Erster Probetest | 2 |
| 2 Testplan | 5 |
| 2.1 1. Testdurchlauf | 7 |
| 2.1.1 Fehlerbereinigung der Probleme aus dem 1. Testdurchlauf | 8 |
| Test-Nr. 47, "Dummyname" für den Alias des Serveranwenders: | 8 |
| Test-Nr. 35, Unterschiede in der Client-Oberfläche gegenüber dem Entwurf | 9 |
| Test-Nr. 44, Server loggt bei Testverbindungen Fehler | 9 |
| Test-Nr. 45, Malbereich nach Initialisierung neumalen | 9 |
| Test-Nr. 46, Server loggt Fehler, beim Datensenden an beendete Clientverbindung | 9 |
| 2.2 2. Testdurchlauf | 10 |
| 2.2.1 Fehlerbereinigung der Probleme aus dem 2. Testdurchlauf | 10 |
| 2.3 3. Testdurchlauf | 11 |

1 Erster Probetest

Auch wenn der erste “erfolgreiche” Start einer Anwendung nicht immer als Test betrachtet werden kann, ist der erste Start mein Testbeginn. Zu Beginn habe ich einen Server mit den Standardeinstellungen gestartet und anschließend drei Clients mit den Farben Rot, Blau und Grün, mit jeweils einem anderen Nutzeralias. Alle vier Portale (Serverkonsole + drei Clients) habe ich nebeneinander gestellt und in den Clients ein paar Probestriche gemalt:



(Leider sind die gemalten Punkte relativ schlecht in der Abbildung zu sehen.)

Unten links in der Ecke ist die Serverkonsole, die drei dunklen Fenster stellen jeweils einen Client da. Was funktioniert noch nicht:

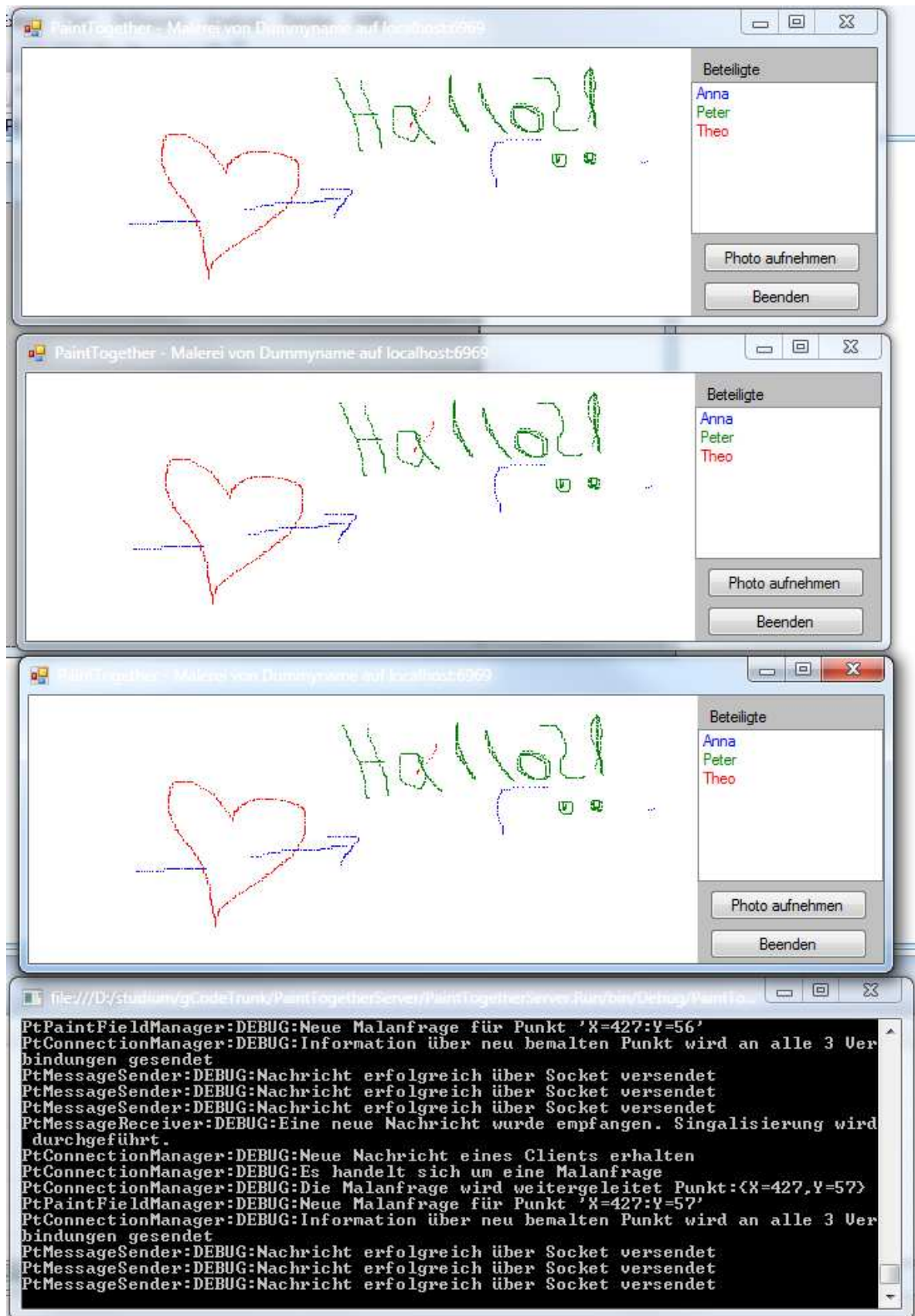
- Bei dem Client rechts unten stehen die drei Alias doppelt drin
- Der Malbereich ist schwarz und nicht weiß (siehe Entwurf)
- Bei dem Client rechts unten ist das gemalte Wort “Grün” fast nur rot, bei den anderen Clients ist das gemalte Wort richtig gefärbt
- Der Malbereich überschneidet die Alias-liste ein Stück
- Die Aliasnamen haben ein zu großen Abstand

Bevor weitere Tests, auch über mehrere Rechner erfolgen, behebe ich ersteinmal die fünf Probleme.

Beim Beheben der Fehler habe ich gleich noch ein paar andere Kleinigkeiten behoben, die mir dabei aufgefallen sind. So konnte man durch Festhalten der linken Maustaste über den

Malbereich hinaus gehen und dann wurde versucht außerhalb des Bildes zu malen. Zudem trat im Client ab und zu ein Fehler in der Malereiverwaltung auf, wenn man einen Punkt bemalt hat und gleichzeitig von einem anderen Beteiligten ein Punkt bemalt wurde. Jetzt wird der Zugriff auf das Bitmap in der FieldManager-EBC des Clientprojekts gelockt.

Nach der ersten Fehlerbeseitigung sieht das ganze schon besser aus:



Unten sieht man die Serverkonsole und oben drei lokale Clients. Neben dem Malen klappt auch das Aufnehmen eines Photos. Leider habe ich bei dem obigen Beispiel kein Photo gemacht, aber hier ein Beispielphoto einer anderen “Gemeinschaftsarbeit” (war bisher nur ich):



Einzig fehlerhaft beim Photo aufnehmen ist die Dateiendung. Das Bild wird als *.bmp abgespeichert, es handelt sich aber um eine *.png. (Wurde jetzt auch behoben)

Nach dem Beenden eines Clients kam auf der Serverkonsole ständig die Nachricht “Nachricht ohne Startblock empfangen”. Das Problem waren die Clientsockets die zwar cleintseitig beendet wurden, aber auf der Serverseite ständig bei Receive() eine leere Nachricht empfangen haben. Ich habe nun eingebaut, wenn ein Socket mehrmals hintereinander eine leere Nachricht empfängt, dass diese Verbindung als “zusammengebrochen” angesehen wird. Zudem wird jetzt auch bei bewusster Beendigung einer Socketüberwachung für den Empfang von Nachrichten im PtMessageReceiver eine “Verbindung verloren” Nachricht geschickt (beschriebenes Problem lag im Communicater). Denn bisher wurde die Beteiligtenliste bei vorheriger Überwachungsbeendigung immer länger. Die Beendigung eines Clients führte bei den anderen Clients nicht zur Reduzierung der Beteiligtenliste.

Nach dem erfolgreichen Start werde ich jetzt die in der Phase 1 verlangte und in Phase 2 entworfene Startanwendung implementieren.

<Implementierungsbeschreibung PaintTogetherStartSelector siehe letzter Abschnitt von Phase 3: Implementierung>

2 Testplan

Nach der erfolgten Erstellung der bisher fehlenden Startanwendung und nachdem der Start eines Servers funktioniert und sich mehrere Clients mit ihm verbinden können, stelle ich einen Testplan auf. Im wesentlichen müssen im Testplan alle in der Analyse/Definition festgelegten Funktionen aufgeführt werden sowie alle zusätzlich im Entwurf definierten Vorgaben. Die folgenden Punkte müssen demnach getestet werden.

Tests für Abdeckung der Anforderungen aus Phase 1 (Analyse/ Definition):

- Startanwendung
 - ☐ Startbildschirm mit Entscheidung für Server oder Clientstart
 - ☐ Eingabe von Farbe und Alias
 - ☐ Bei Entscheidung für Server
 - ☐Eingabe der Größe
 - ☐ Bei Entscheidung für Client
 - ☐Eingabe IP oder Rechnername (Std. localhost) und Port (Std. 6969)
- Verbindungen
 - ☐ 10 PT-Clients mit einem Server verbinden und die Malfunktion testen
 - ☐ Starten von 2 PT-Servern auf einem Rechner mit verschiedenen Ports
 - ☐ Verbindung von einem Rechner mit 2 PT-Servern auf anderen Rechnern
 - ☐ Start des PT-Servers mit der Maximalgröße von 1000*500
 - ☐ Start des PT-Servers mit der Minimalgröße von 500*200
 - ☐ Start des PT-Servers mit einer Malgröße außerhalb der min-max-Beschränkung
 - ☐ PT-Server und PT-Client auf WindowsXP und Windows 7 starten
 - ☐ PT-Server auf WindowsXP und PT-Client auf Windows 7 starten
 - ☐ PT-Server auf Windows 7 und PT-Client auf WindowsXP starten
 - ☐ 2 Clientverbindungen mit identischem Namen und identischer Farbe zu einem Server
- PT-Client
 - ☐ Verarbeitung der Startparameter (Alias, Farbe, Zielrechner, Zielport)
 - ☐ Beim Start soll Größe und Malinhalt vom Server übernommen werden
 - ☐ Photo aufnehmen und mit der Malerei vergleichen
 - ☐ Wird beim Malen auch mit der vorher eingestellten Farbe gemalt?
 - ☐ Wird auch übermalt?
 - ☐ Malvorgang eines Clients muss auf anderen Clients ebenfalls erfolgen
 - ☐ Anzeige aller gerade beteiligter Personen mit Alias und Farbe
- PT-Server
 - ☐ Verarbeitung der Startparameter (Alias, Leinwandgröße)
 - ☐ Konsole für Logging
 - ☐Client-Verbindungs- Auf- und Abbau
 - ☐Malaktion durch einen Client

Tests für Abdeckung der Anforderungen aus Phase 2 (Entwurf):

- Startanwendung
 - ☐ Startbildschirm
 - ☐Gestaltung wie im Entwurf?
 - ☐Verhalten von “Farbe wählen” wie im Entwurf?
 - ☐Verhalten der Aktionsbuttons wie im Entwurf?

- ☐ Bildschirm “Malereieinstellungen”
 - ☐ Gestaltung wie im Entwurf?
 - ☐ Standardwerte vorausgefüllt?
 - ☐ Verhalten der Aktionsbuttons wie im Entwurf?
 - ☐ Grenzwertprüfung für Malereigrößenangabe?
- ☐ Bildschirm “Malereisuche”
 - ☐ Gestaltung wie im Entwurf?
 - ☐ Standardwerte vorausgefüllt?
 - ☐ Verhalten der Aktionsbuttons wie im Entwurf?
- PT-Client
 - ☐ Gestaltung wie im Entwurf?

Tests für Abdeckung der Anforderungen aus Phase 3 (Implementierung):

- Startanwendung
 - ☐ Aliasfelder -> Wertepfung auf von mir festgelegte ungültige Zeichen
 - ☐ Portfelder -> Prüfung das Port größer als 1000 und kleiner, gleich 99999 ist
- PT-Server und PT-Client
 - ☐ Startparameter Alias -> Wertepfung wie bei Startanwendung
 - ☐ Startparameter Serverport -> Wertepfung wie bei Startanwendung
 - ☐ Startparameter Farbe bei PT-Client kann Farbname oder “R-G-B” verstehen

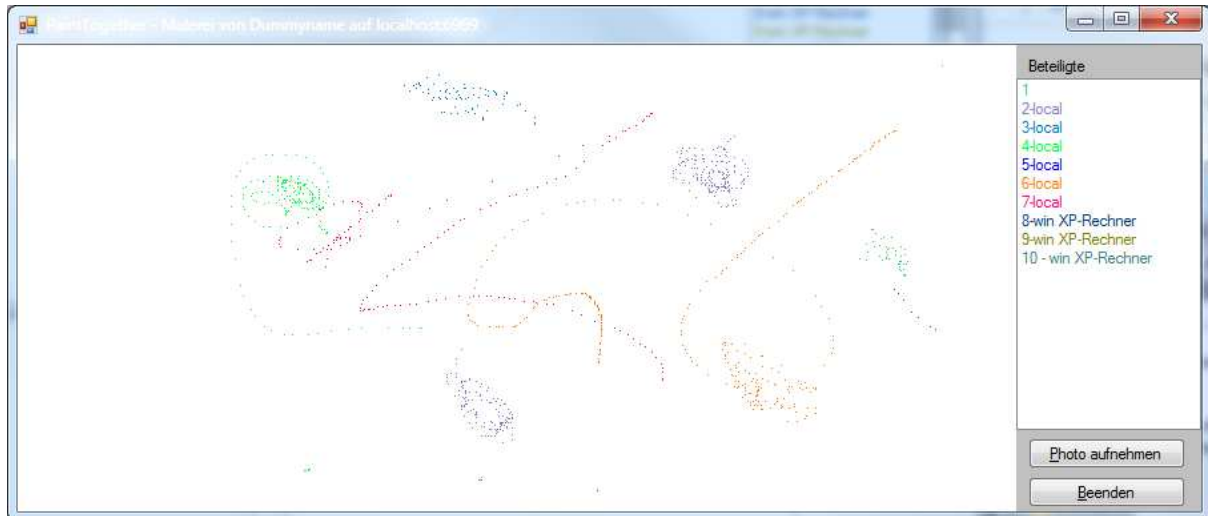
Für die hier aufgelisteten vorzunehmenden Tests erstelle ich eine Test-Tabelle.

<siehe Dokument “Phase 4.2: Testplan”>

2.1 1. Testdurchlauf

Die Tests aus dem aufgestellten Testplan habe ich alle abgearbeitet. Dabei ist bei einem Test eine Unstimmigkeit zu dem Entwurf aufgetreten. Der Test-Nr. 35: “Client: Gestaltung wie im Entwurf?” ergab die Antwort Nein, weil siehe Bemerkung “Im Entwurf gibt es kein ICON - aktuell gibts das Windowsstandard-ICON, sollte ausgeblendet werden + Maximierenbutton ist im Entwurf nicht vorhanden - sollte ausgeblendet werden”. Dies ist der einzige Test aus dem Testplan, der nicht zufriedenstellend durchgeführt werden konnte.

Screenshot für Test-Nr. 18: “Erster Testdurchlauf 10 Verbindungen”:



Es sind mir bei dem 1. Testdurchlauf noch vier zu beseitigende, nicht durch den Testplan abgedeckte Probleme aufgefallen. Diese habe ich nachträglich in den Testplan mit aufgenommen. Es handelt sich um den letzten Bereich in der Tabelle “Sonstiges während des Tests aufgefallen”.

2.1.1 Fehlerbereinigung der Probleme aus dem 1. Testdurchlauf

Die nunmehr fünf offenen Probleme (Test-Nr: 35, 44, 45, 46, 47), die im 1. Testdurchlauf gefunden wurden, behebe ich jetzt, anschließend wird ein 2. Testdurchlauf durchgeführt, der natürlich auch die schon erfolgreich getesteten Tests der geänderten Komponenten mit einbeziehen muss, da Änderungen sich auch auf diese Funktionen auswirken könnten. Ändere ich zum Beispiel etwas am PaintTogetherCommunicater, dann muss ich alle Tests des Clients und Servers erneut durchführen, ebenso die Verbindungstests. Nur die Startanwendung brauche ich in dem Fall nicht noch einmal testen.

Test-Nr. 47, “Dummyname” für den Alias des Serveranwenders:

Zwar wurde der Alias schon als Parameter im “Run-Projekt” eingelesen, aber beim Start des Servers nicht verwendet. Die PtConnectionManagerEBC benötigte den Servernamen als Initialisierungs-Input-Pin. Diesen habe ich hinzugefügt (ProcessInitConnectionManagerMessage). Da es sich aber um eine innere EBC der PtServerClientAdapter-Platine handelt und die Platine schon den Inputpin “ProcessStartPortListingMessage” besaß, habe ich einfach den Inputpin nach “ProcessInitAdapterMessage” umbenannt und die Nachrichtenklasse um einen Alias erweitert. Die Platine empfängt jetzt wie bisher auf dem Inputpin das Nachrichtenobjekt, aber leitet dieses jetzt nicht wie üblich nur an eine der internen EBCs, sondern an zwei weiter. Aus der InitApdaterMessage erstellt die Platine zwei Nachrichten, eine wie bisher für den PtPortListener

mit dem zu überwachenden Port und die neue Nachricht für den PtConnectionManager mit dem Alias des Servernutzers. Die PtClients sollten jetzt bei erfolgreichem Verbindungsaufbau den richtigen Alias erhalten.

Zusätzlich habe ich die ändernden Zugriffe auf die Verbindungslisten im ConnectionManager mit einem lock-Objekt “geschützt”. Bei vielen gleichzeitigen An- und Abmeldungen konnte es hier zu einem Fehler führen. Der Fehler ist bei dem Durchlauf der bestehenden Testcases aufgefallen.

Test-Nr. 35, Unterschiede in der Client-Oberfläche gegenüber dem Entwurf

Wie in der Beschreibung des 1. Testdurchlaufs zu sehen soll noch das Windowsstandardapplikationicon sowie der Maximierbutton ausgeblendet werden. Für diese Anpassung musste ich lediglich zwei Einstellungen im Formdesigner des Clientportals ändern.

Test-Nr. 44, Server loggt bei Testverbindungen Fehler

An eine neue Clientverbindung werden jetzt erst die Initialisierungsinformationen geschickt, wenn von der Clientverbindung die Connected-Nachricht empfangen wurde. Bei Testverbindungen, wo nur ein Socket geöffnet und wieder geschlossen wird, sollte kein Fehler mehr beim Senden auftreten, da keine Daten mehr an diese Verbindungen geschickt werden.

Test-Nr. 45, Malbereich nach Initialisierung neumalen

Für die Behebung dieses Fehlers musste einfach nach der Initialisierung des Malpanels die GUI zum Neumalen veranlasst werden. (Aufruf von Invalidate() am Control) Die Änderung erfolgte in der internen EBC “PtConnectionManager” des PtServerClientAdapters im PaintTogether-Haupt-Projekt.

Test-Nr. 46, Server loggt Fehler, beim Datensenden an beendete Clientverbindung

Keine Änderung für diesen Fehler am PaintTogehterServer vorgenommen. Bei dem Senden von Informationen werden immer alle gerade als bestätigt markierten Verbindungen verwendet. Wenn während des Sendevorgangs eine Verbindung getrennt wird, an die noch keine Information geschickt wurde, dann gibt es einen Fehler beim Senden. Der Fehler beschränkt sich aber auf die getrennte Verbindung. Als Fehlerbehebung könnte ich nicht einmal beim Senden die Liste mit den gerade verbundenen Socket sperren, da die Verbindung ja clientseitig getrennt wurde. Ich denke das Loggen des Fehlers ist in diesem Fall richtig bzw. lediglich unschön. Ich lasse diese “Gefundene Unschönheit” also mit Absicht so wie sie ist und starte den 2. Testdurchlauf.

2.2 2. Testdurchlauf

Da an der Startanwendung und der Parameterauswertung von Server und Client nichts geändert wurde, führe ich die Tests der Startanwendung nicht noch einmal durch. Ebenso lasse ich die Tests für die Auswertung der Startparameter an Server und Client aus. Alle anderen Tests müssen im 2. Testdurchlauf noch einmal durchgeführt werden.

Beim zweiten Testdurchlauf sind glücklicher Weise keine weiteren Probleme aufgetaucht.

<siehe Dokument “Phase 4.2: Testplan - 2. Testdurchlauf”>

Auch alle im 1. Testdurchlauf gefundenen Probleme wurden beseitigt. Bei der Unschönheit Test-Nr. 46 konnte ich die Fehlerlognachrichten nicht erneut nachstellen. Ggf. ist das Problem durch Änderungen für Test-Nr. 46 mit dem locken der Zugriffe auf die Socketverbindungslisten “verbessert” worden.

Screenshot für Test-Nr. 18: “2ter Testdurchlauf 10 Verbindungen”:



Allerdings ist mir jetzt aufgefallen, dass meine “Striche” Pünktchenlinien sind. Ich habe noch einmal in die Anforderungen geschaut und da habe ich geschrieben, dass der “Stift” sich wie der Stift in Paint verhalten soll. In Paint werden jeweils zwei Punkte mit einem durchgehenden Strich verbunden. Also werde ich jetzt noch diese Funktionalität als Test-Nr: 48 in den Testplan mit aufnehmen und beheben.

2.2.1 Fehlerbereinigung der Probleme aus dem 2. Testdurchlauf

Hierbei ist nur das Problem Test-Nr. 48 zu sehen. Für die Umsetzung werde ich keine neuen In- und Outputpins an den EBCs definieren. Ich verwende die aktuelle Malnachricht, wo ein Punkt übertragen wird. Jetzt sollen zwei Punkte übertragen werden, Server und Client müssen dann selbst auf ihren Bitmaps eine Linie mit der angegebenen Farbe zwischen den beiden Punkten malen.

Jetzt werden wirklich Striche und keine einzelnen Punkte mehr gemalt. Der “Stift” funktioniert nun wie in den Anforderungen definiert. Da ich sowohl im Communicater als auch im Client und Server Änderungen vorgenommen hat, muss ich einen dritten Testdurchlauf durchführen.

2.3 3. Testdurchlauf

<siehe Dokument “Phase 4.2: Testplan - 3. Testdurchlauf”>

Screenshot für Test-Nr. 18: “3ter Testdurchlauf 10 Verbindungen”:



Notiz:

Die letzte Phase des Projektes ist somit abgeschlossen. Eine ausführbare Anwendung werde ich bei dem Googlecode-Projekt als Download bereitstellen sowie bei der Aufgabeneinsendung mitschicken. In einem abschließenden Dokument “Phase 5: Wartung und Pflege” werde ich noch kurz Möglichkeiten zur Weiterentwicklung und einen Rückblick zur Durchführung des Projekts geben.