

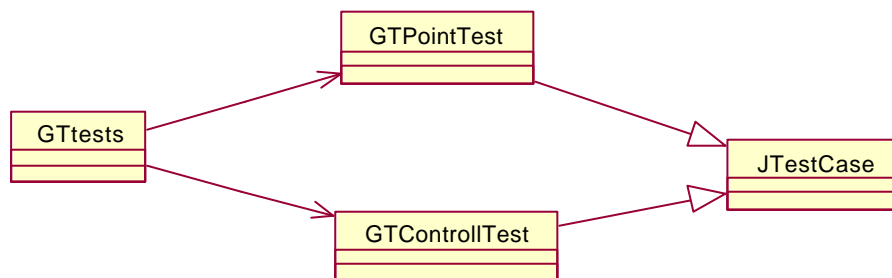
Testkonzept

Einführung

Es werden zwei verschiedene Arten von Tests benutzt: Inspektiv und durch JUnit. Die inspektive Variante wird zum Beispiel bei dem Viewpaket die einzige sein. Entweder werden die Fenster richtig erschafft und die Toolbars und Menubars richtig angelegt, oder nicht. Viele andere Möglichkeiten gibt es nicht um festzustellen, ob es richtig ist oder nicht. Für diese Variante sollten vom jeweiligen Team Checklisten vorbereitet werden, welche abgearbeitet werden müssen.

Anders ist es beim Modelpaket oder beim Controllpaket, die intensives testing durch das JUnit Framework erfordern. Jedes Team muss ein entsprechendes Dokument (**Testkonzept**) erstellen, welches allgemein die Testkonzepte und die Notwendigkeit dieser erklärt. Danach müssten diese durch JUnit implementiert werden. Das Testkonzept sollte schon vor dem Programmieren fertig sein, während die JUnit Klassen und Methoden simultan zum Programmieren erstellt werden sollten.

Jedes Team muss seine eigenen TestCase(s) entwerfen, die zu durchlaufen sind bevor der Quellcode im CVS abgelegt wird. Bevor das Projekt dem Tutor vorgestellt wird, werden alle Tests zusammengefasst und am ganzen Projekt ausgeführt.



Testkonzept vom Viewpaket

Wie schon oben beschrieben wird dieser eher inspektiv durchgeführt.

Testkonzept vom Modelpaket

- Es muss getestet werden, ob eingefügte Punkte auch wirklich in der Konfiguration existieren. Der Folgende PseudoCode zeigt ein Beispiel wie dies realisiert wird:

```
public class GTPointTest extends TestCase{
    ...
    public void setUp(){
        //neue Konfiguration starten
    }
    public void TestAddPoint(){
        GChildFrame active = GMainFrame.getActiveWindow();
        Point2D.DoublePoint P = new Point(100,200);
        active.mouseClicked(MouseEventAtPointP);
        GPoint p = GMainFrame.getActiveConfiguration().findPointAt(P);
        assertTrue(p!=null);
        assertTrue(p.getDoublePoint()==P);
    }
    ...
}
```

- Es muss getestet werden, ob ein Punkt nach einer spezifischen Verschiebung auch entsprechend in der Konfiguration modifiziert wird. Das Beispiel dafür wäre ähnlich wie oben: Punkt an Koordinaten (X,Y) hinzufügen, mouseDown am Punkt, mouseMove, mouseReleased am Punkt (X2,Y2), und dann sicherstellen das der Punkt der hinzugefügt wurde auch an den Koordinaten (X2,Y2) ist.

Testkonzept vom Controlpaket

Eine Klasse GTestControl:JTestCase sollte entworfen werden, die überprüft ob nach Erschaffung eines GChildWindows dieses auch in der WindowListe von GMainFrame existiert, ob Ansichten erschaffen werden, und alle möglichen Varianten von Fenstern schliessen, erschaffen und skalieren möglich sind. Weiterhin soll überprüft werden, ob das Gitter ein-/ausgeblendet wird. Dasselbe auch für die Koordinatenachsen.

Die Feinheiten des Testkonzepts werden erst nach der ersten Iterationsrunde eindeutig werden und eine sichere Grundlage für weitere Versionen darstellen.