
SmaSTra Development Report



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Elektrotechnik
und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation
Prof. Dr.-Ing. Ralf Steinmetz

SmaSTra Development Report

Technische Universität Darmstadt
Fachbereich Elektrotechnik und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation (KOM)
Prof. Dr.-Ing. Ralf Steinmetz

Contents



1 SmaStra Entwicklungsverlauf

1.1 Stand der Alpha

TODO: Entwicklung von Smastra von Begin bis zur Präsentation der ersten Alpha Version

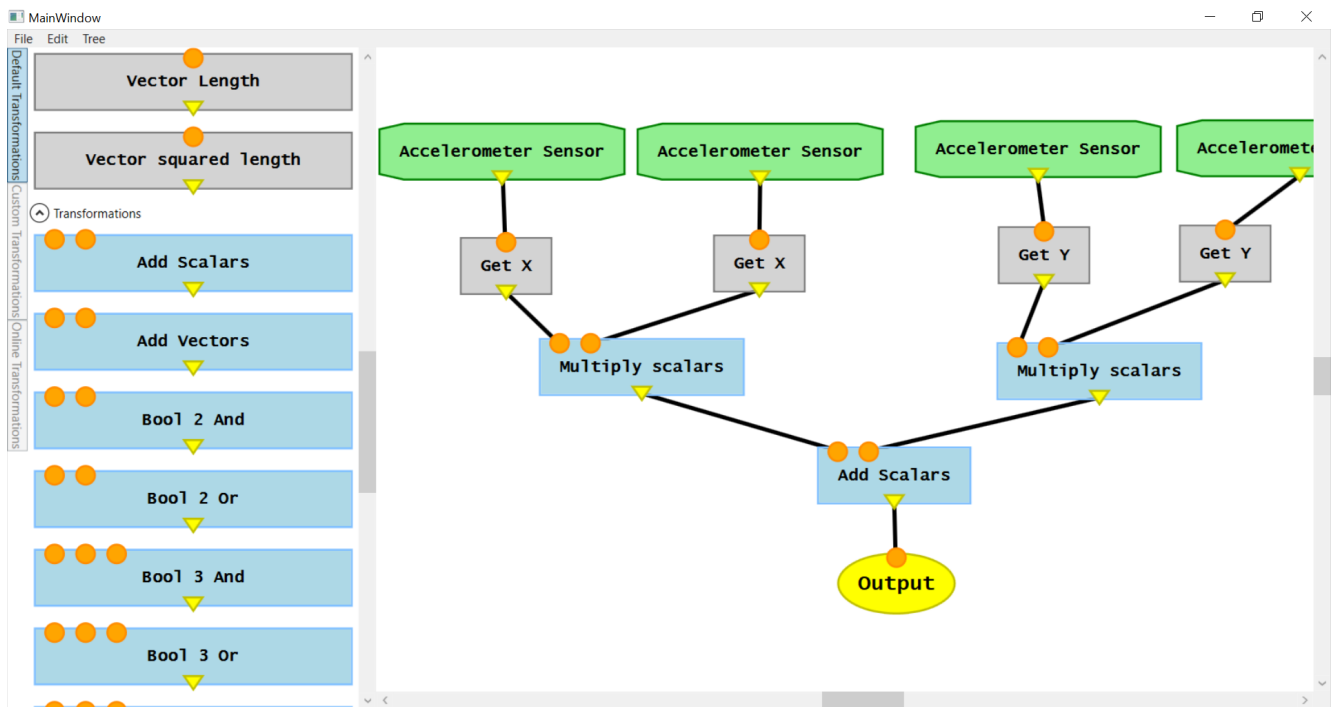


Figure 1.1: Stand von SmaSTra in der Alpha Version

- welche Features sollten in der Alpha Version umgesetzt werden
- welche Features wurden tatsächlich umgesetzt / was konnte die Alpha Version
- Feedback/Evaluation/Kritikpunkte der Alpha Version

1.2 Weiterentwicklung der Alpha

Die Alpha Version zeigte, dass das System umsetzbar war. Allerdings wurden auch etliche Mängel deutlich, die den Umgang mit dem Tool erschwerten. Unübersichtlichkeit, fehlende Funktionalitäten und fehlendes visuelles Feedback waren mit die größten Kritikpunkte, die im Laufe der weiteren Entwicklung von SmaSTra behoben werden mussten.

1.2.1 Neue Aufteilung der Nutzeroberfläche

Bei der Auswertung der Alpha Version zeigte sich, dass die bisherige Aufteilung der Nutzeroberfläche den Anforderungen nicht gerecht wird. Daher wurden einige grundlegende Änderungen eingeführt (vgl. Figure ??). Um einen besseren Überblick der zur Verfügung stehenden Elemente bieten zu können, wurde die Elementauswahl in den gesamten oberen Bereich der Oberfläche verlegt. Entlang der rechten Seite wurden die zwei neue Bereiche Properties (oben) und Library (unten) hinzugefügt. Im Properties

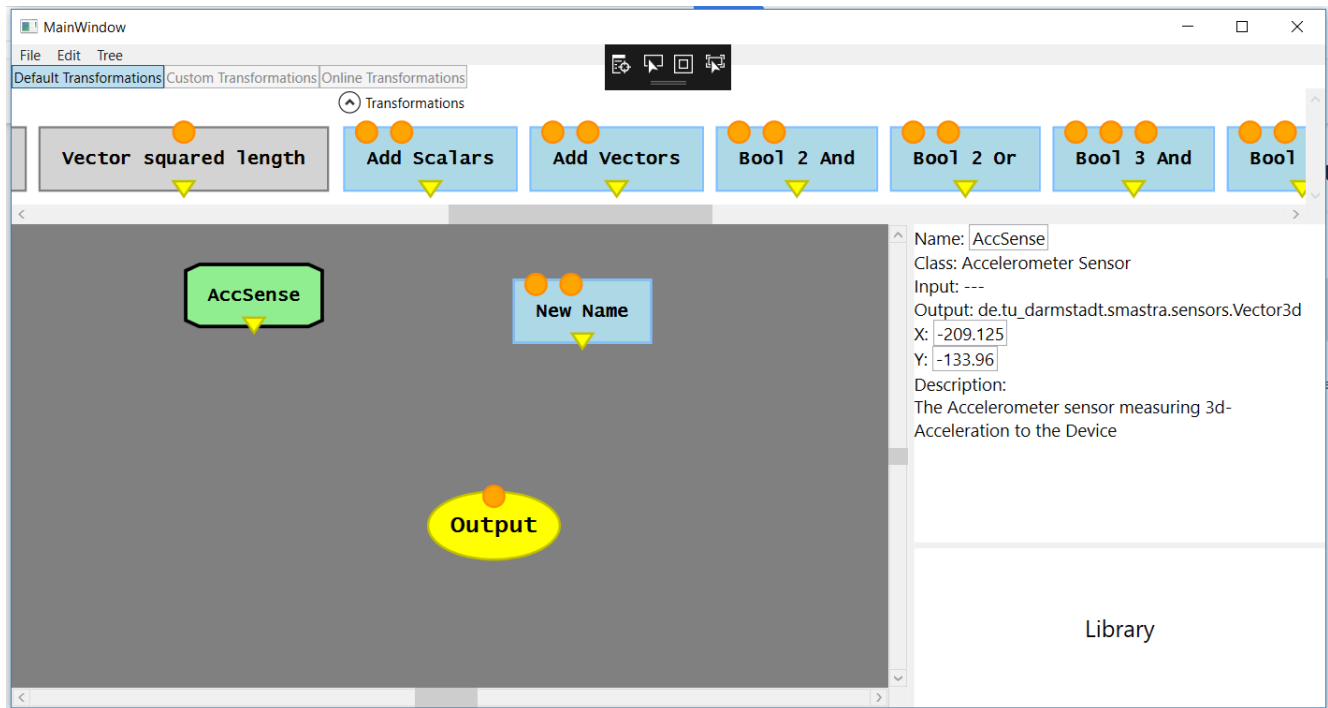


Figure 1.2: Aufteilung der GUI in Arbeitsfläche, Obere Auswahlleiste, Properties und Library

Bereich können die Details des aktuell ausgewählten Elements bearbeitet werden. In der Library können Elemente gespeichert werden, um so einen schnellen Zugriff auf häufig verwendete Elemente zu gewährleisten. Die gesamte restliche Fläche wird als Hauptarbeitsbereich verwendet.

1.2.2 Neue Repräsentation der Elemente

Nachdem der grundlegende Aufbau der neuen Oberfläche umgesetzt worden war, stand eine Überarbeitung der Elemente selbst an oberster Stelle (vgl. Figure ??).

Anstatt an der Ober- und Unterseite, sind die Sockets zur Ein- und Ausgabe nun seitlich an den Elementen angebracht. Dadurch verläuft die allgemeine Flussrichtung horizontal, wodurch die verfügbare Fläche des Hauptarbeitsbereiches besser genutzt werden kann. Zudem steht dadurch neben den Eingabesockets zusätzlicher Platz zur Verfügung. Dieser wird genutzt, um den erwarteten Datentyp eines Eingabesockets anzuzeigen, solange an diesen noch kein Wert angelegt wurde. Sobald eine Verbindung an dem entsprechenden Socket angelegt oder ein Wert per Hand im Properties Bereich eingetragen wurde, wird, anstatt des Datentyps, der Name des verbundenen Elements bzw. der eingetragene Wert angezeigt.

Sämtlichen Datentypen, die von den vorinstallierten Elementen genutzt werden, wurde jeweils eine bestimmte Farbe zugeordnet und alle Eingabe- und Ausgabesockets werden in der Farbe ihrer entsprechenden Datentypen dargestellt. Auf diese Weise kann der Nutzer auf den ersten Blick sehen, um welche Datentypen es sich handelt und ob diese miteinander verbunden werden können.

1.2.3 Auswahlleiste und Menüs

Der Hintergrund der gesamten Nutzeroberfläche wurde abgedunkelt. Dadurch stechen die farbigen Elemente markanter hervor und lenken somit die Aufmerksamkeit des Nutzers auf die wichtigeren Bere-

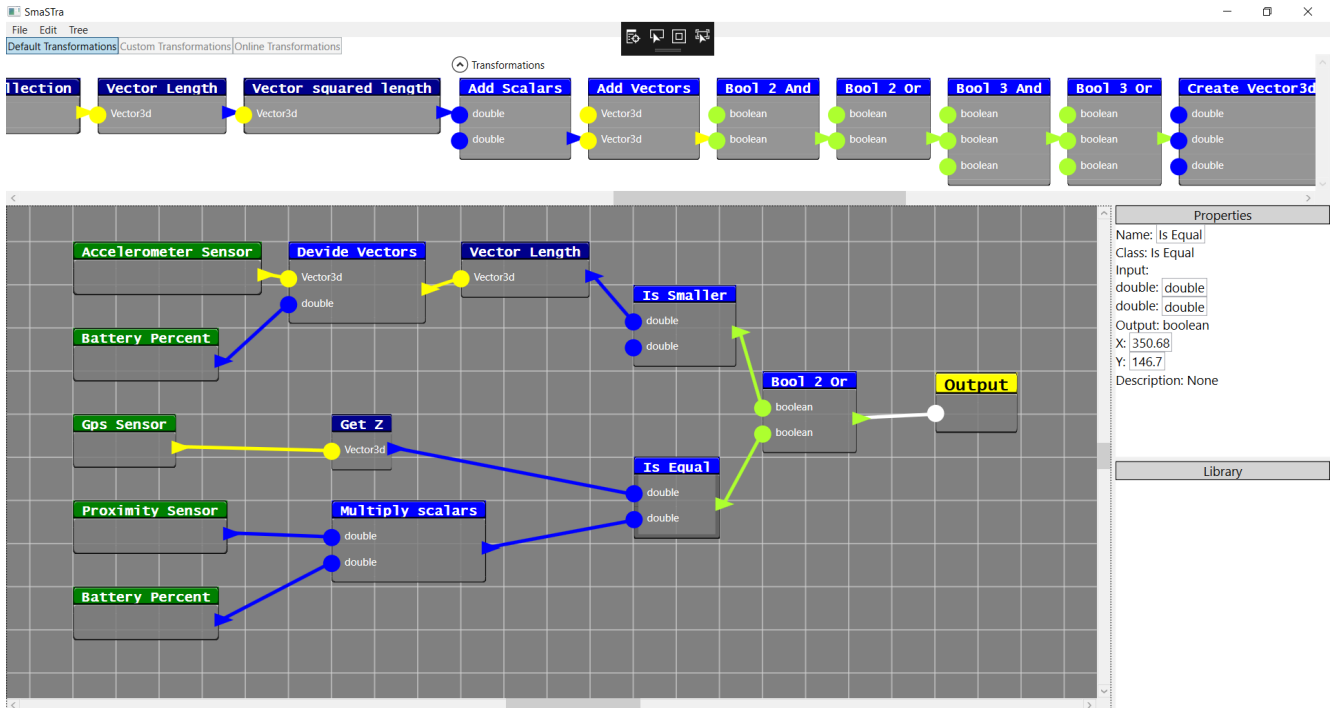


Figure 1.3: Die Elemente wurden komplett überarbeitet: Text und Farbe liefern Informationen zu den Eingabe- und Ausgabesockets. Der Arbeitsverlauf ist nun horizontal.

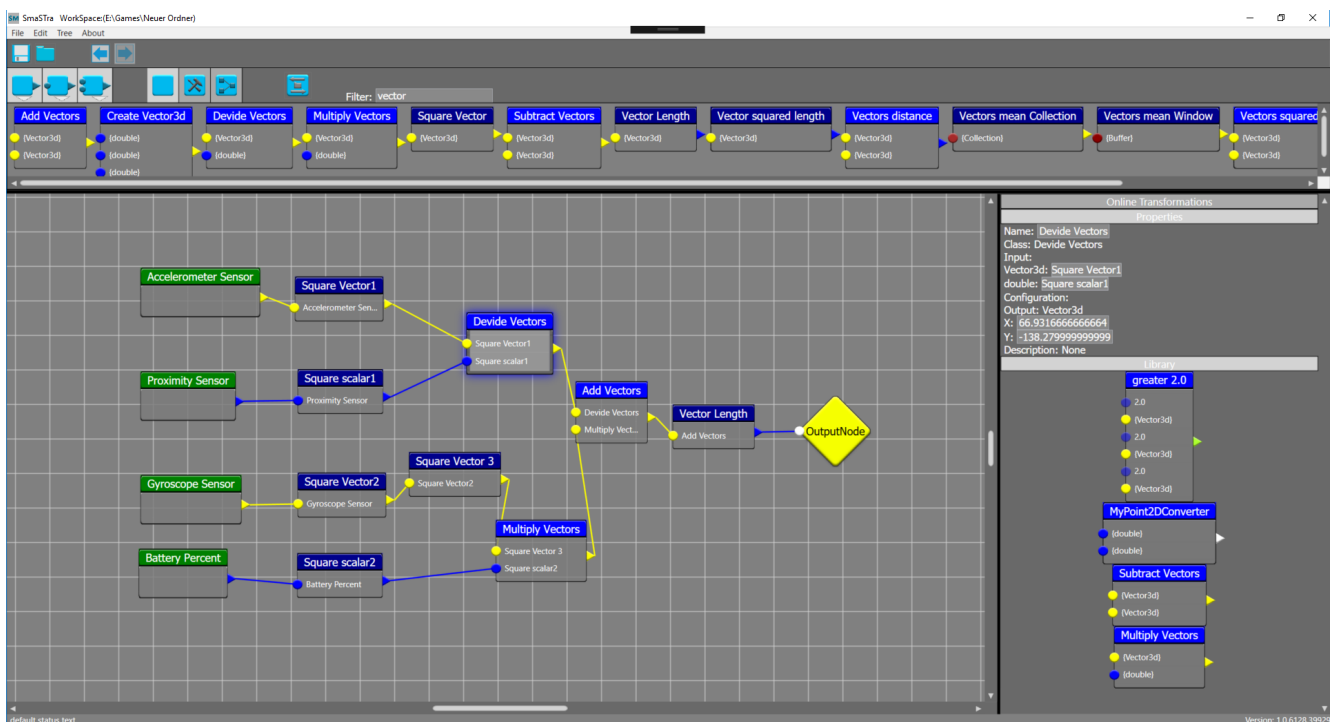


Figure 1.4: Die Auswahlleiste verfügt nun über mehrere Filteroptionen um bestimmte Elemente schneller zu finden.

iche. Zudem ist das gewählte Farbthema bei längeren Arbeiten schonender für die Augen als eine grelle Arbeitsoberfläche.

Zur effektiveren Handhabung der Vielzahl an unterschiedlichen Elementen, wurden weitere Kategorien eingeführt nach denen die Elemente gefiltert werden können. Diese Kategorien unterscheiden

zum einen nach der Anzahl der Eingabewerte und zum anderen danach, ob das Element von Nutzern erstellt wurde oder bereits vorinstalliert war. Zudem erlaubt ein Eingabefeld das Filtern nach beliebigen Stichwörtern, um so schnell Elemente aufzulisten, die thematisch zusammen gehören.

Diese Filteroptionen wurden zusammen mit häufig verwendeten Funktionen in einer Menüleiste entlang des oberen Randes der Nutzeroberfläche platziert, so dass sie jederzeit sichtbar und schnell erreichbar sind. Außerdem wurden eine Reihe von Tastenkürzeln und Kontextmenüs eingeführt, die dem Nutzer das Navigieren über die Arbeitsfläche und das Arbeiten mit den Elementen erleichtert.

2 SmaSTra Bedienungsanleitung

2.1 Was ist SmaSTra?

2.2 Beispielanwendung

In diesem Abschnitt werden die grundlegenden Funktionen von SmaSTra anhand eines einfachen Anwendungsbeispiels gezeigt. In diesem Beispiel wird ein Sensor erstellt, der ausgeben soll, ob ein Gerät (beispielsweise ein Smartphone) schnell in eine beliebige Richtung beschleunigt wird. Zu diesem Zweck soll überprüft werden, ob einer der Ausgabewerte des "Linear Accelerometer Sensors" einen bestimmten Schwellenwert überschreitet.

Elemente auswählen

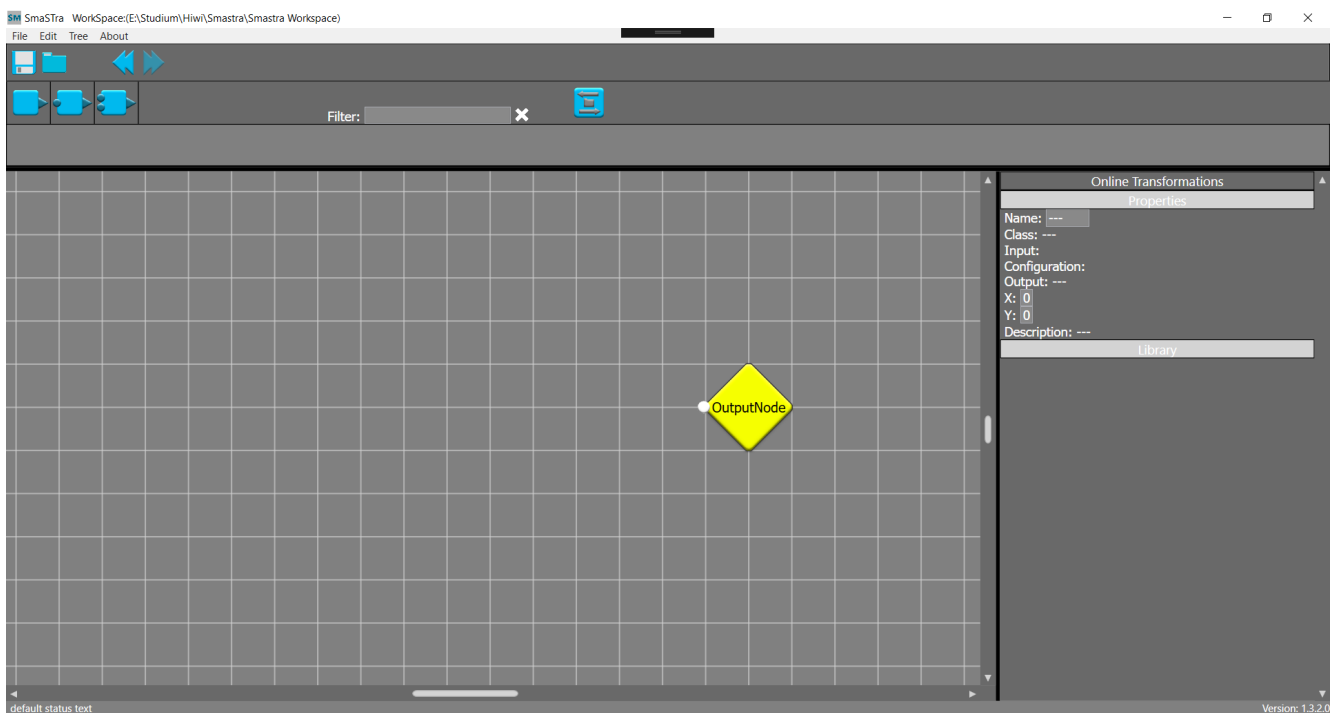




Figure 2.1: Der Hauptbildschirm direkt nach dem Start von SmaSTra

Nach dem Starten von SmaSTra und der Anzeige des Splashscreen, erscheint der Hauptbildschirm (vgl. Figure ??). Im Arbeitsbereich befindet sich nur das Output-Element.

Nun soll zunächst ein "Linear Accelerometer Sensor" in den Arbeitsbereich gezogen werden. Mit einem Klick auf die Schaltfläche "Data Sources" , werden diejenigen vorinstallierten Elemente angezeigt, die als Datenquellen dienen. Per Drag&Drop kann der Linear Accelerometer Sensor im Arbeitsbereich platziert werden, am besten in der Nähe des linken Randes.

Anhand der Farbe und des Tooltips des Ausgabesockets kann man ablesen, dass der Ausgabebetyp des Linear Accelerometer Sensor ein 3-dimensionaler Vektor ist. Jede Komponente des Vektors gibt die Beschleunigung in eine bestimmte Richtung an (vor/zurück, links/rechts, hoch/runter). Um mit dem Wert einer einzelnen Komponente des Vektors arbeiten zu können, wird ein weiteres Element benötigt. Mit einem Klick auf die Schaltfläche "Conversions"  werden auch diejenigen Elemente angezeigt, die

einen einzelnen Eingabewert erwarten. Suchen Sie das Element "Get X" und platzieren Sie es rechts neben dem Linear Accelerometer Sensor. Wie man an der Farbe des Eingabesocket von Get X erkennen kann, wird hier ein 3-dimensionaler Vektor erwartet. Der Ausgabesocket des Linear Accelerometer Sensors kann also mit dem Eingabesocket von Get X verbunden werden.

Elemente Verbinden

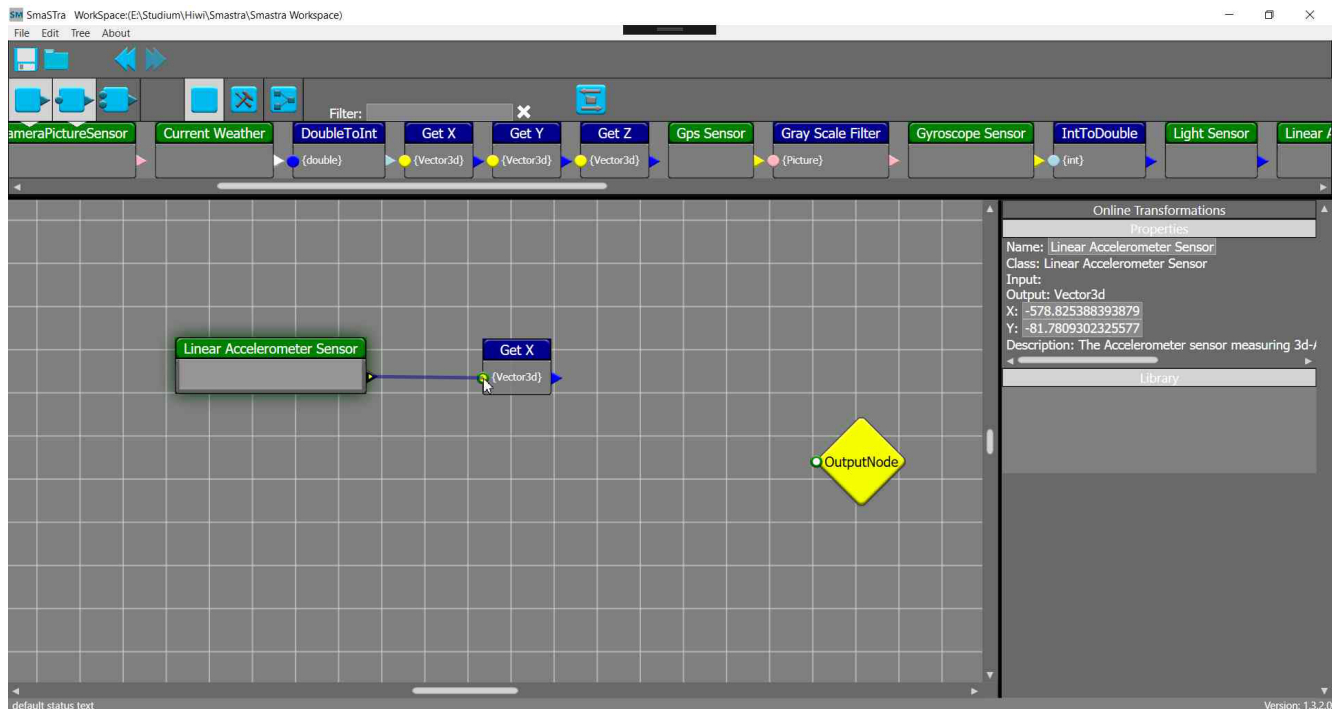


Figure 2.2: Verbinden des Linear Accelerometer Sensors mit Get X

Um eine Verbindung zwischen zwei kompatiblen Elementen herzustellen, klicken und halten sie einen der beiden Sockets gedrückt und bewegen Sie die Maus bei gedrückter Maustaste. Eine dunkelblaue Verbindungslinie erscheint zwischen ihrer Maus und dem gehaltenen Socket. Außerdem erscheint um sämtliche kompatiblen Sockets ein grüner Rand, während inkompatible Sockets rot umrandet werden (vgl. Figure ??). Bewegen Sie nun die Maus über den Socket mit dem Sie eine Verbindung aufbauen möchten und lassen Sie die Maustaste los. Eine Linie verbindet nun dauerhaft die beiden Elemente. Die Farbe der Linie entspricht dabei der Farbe der Datentypen der Sockets. Zusätzlich wird neben dem Eingabesocket der Name des Verbundenen Elements angezeigt (in diesem Fall Linear Accelerometer Sensor). Die Ausgabe von Get X ist vom Typ double und entspricht in diesem Fall der Beschleunigung des Smartphones entlang der X-Achse.

Werte per Hand eintragen

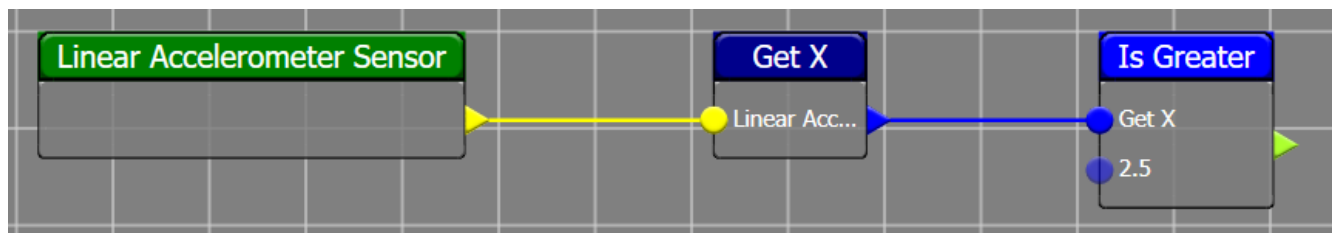


Figure 2.3: Der Wert des unteren Eingabesockets von Is Greater wurde per Hand auf 2.5 gesetzt

Als nächstes soll verglichen werden, ob die Beschleunigung entlang der X-Achse einen bestimmten Schwellenwert überschreitet. Für dieses Beispiel wählen wir einen Wert von 2.5m/s^2 . Für den Vergleich wird das Element "Is Greater" verwendet. Mit einem Klick auf die Schaltfläche "Transformations" werden auch diejenigen Elemente angezeigt, die mehr als einen Eingabewert erwarten. Ziehen Sie Is Greater in den Arbeitsbereich rechts neben Get X und verbinden Sie den Ausgabesocket von Get X mit dem oberen Eingabesocket von Is Greater. Da wir den Schwellenwert selbst auf 2.5 festlegen und er von keinen anderen Elementen abhängt, können wir diesen Wert per Hand eintragen. Mit einem Doppelklick auf den unteren Eingabesocket von Is Greater werden die Details dieses Elements im "Properties" Bereich am rechten Rand angezeigt. Unter "Inputs" werden für beide Eingabesockets jeweils der erwartete Datentyp (double) und der aktuell angelegte Wert angegeben. Als Wert des oberen Sockets wird der Name des verbundenen Elements angezeigt (Get X). Der Wert des unteren Sockets ist noch leer, da noch kein Wert angegeben wurde. In dieses leere Feld kann nun der Wert von 2.5 eingetragen werden. Durch das Eintragen eines Wertes per Hand wird im Is Greater Element der entsprechende Eingabesocket transparent und der entsprechende Wert erscheint neben dem Socket (vgl. Figure ??).

Kopieren und Filtern von Elementen

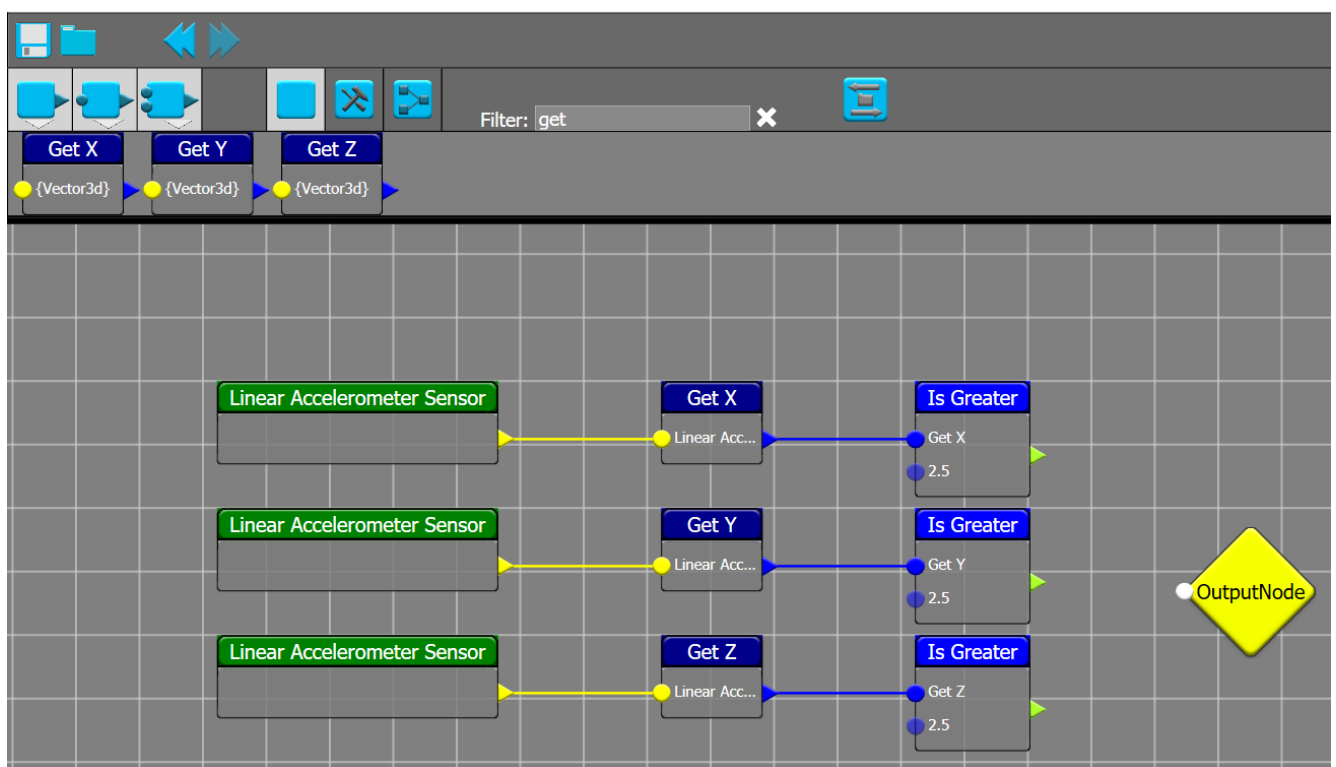


Figure 2.4: Weitere Linear Accelerometer Sensoren und Is Greater Elemente wurden durch Kopieren hinzugefügt. Die Auswahlliste wird nach "get" gefiltert

Nun sollen auch Bewegungen in Y- und Z-Richtung erkannt werden. Zu diesem Zweck werden je zwei weitere Linear Accelerometer Sensoren und Is Greater Elemente benötigt. Anstatt diese Elemente aus der oberen Auswahlliste heraus zu ziehen, können Elemente, die sich bereits im Arbeitsbereich befinden, auch direkt kopiert werden. Wählen Sie den Linear Accelerometer Sensor im Arbeitsbereich mit einem Linksklick aus. Fügen Sie dann Is Greater mit Strg+Linksklick zur Auswahl hinzu. Mit Strg+V wird nun von beiden Elementen jeweils eine Kopie erstellt. Ziehen Sie die Kopien ein Stück nach unten und kopieren Sie sie erneut. Der in Is Greater eingetragene Wert bleibt dabei erhalten. Als nächstes werden die zwei Elemente Get Y und Get Z benötigt. Die Auswahlliste dürfte inzwischen eine Vielzahl unterschiedlicher Elemente anzeigen. Um Get Y/Z schnell zu finden, kann der Filter verwendet werden.

Geben Sie in das Eingabefeld des Filters "Get" ein, um alle Elemente anzuzeigen, die diese Zeichenfolge in ihrem Namen haben (vgl. Figure??). Stellen Sie sicher, dass dabei die Schaltfläche für Conversions aktiv ist. Nun können Get Y und Get Z in die Arbeitsfläche gezogen und analog zu Get X mit den Kopien des Linear Accelerometer Sensors und Is Greater verbunden werden. Leeren Sie den Filter anschließend wieder, um alle Elemente anzeigen zu lassen.

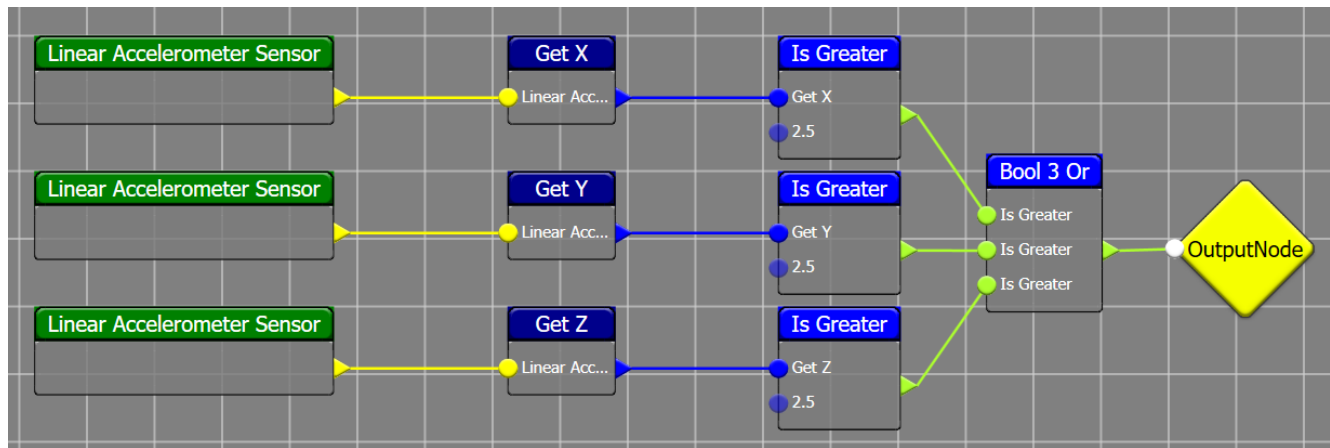


Figure 2.5: Alle Elemente sind durch Bool 3 Or mit dem Output verbunden. Der Java Code kann generiert werden

Abschließend müssen die drei erstellten Zweige noch korrekt zusammengeführt werden. Da als Ergebnis "true" ausgegeben werden soll, wenn einer der Zweige "true" ergibt, müssen diese mit einer Or-Verknüpfung verbunden werden. In den Transformations findet sich dafür das Element "Bool 3 Or". Ziehen Sie es in den Arbeitsbereich links neben Output. Verbinden Sie die Ausgabesockets der Is Greater Elemente mit jeweils einem Eingabesocket von Bool 3 Or und dessen Ausgabesocket mit dem Output. Der Arbeitsbereich sollte aktuell ähnlich aussehen wie in Figure ??.

Erstellen einer Java Klasse

Nun ist es an der Zeit für den ersten Testlauf. In diesem Anwendungsbeispiel, wird davon ausgegangen, dass der von SmaSTra generierte Code in einem Android Studio Projekt verwendet werden soll. Eine Beschreibung zur Exportierung des Codes in andere Entwicklungsumgebungen, wie zum Beispiel Eclipse, findet sich in Abschnitt ??.

Erstellen Sie ein neues Android Studio Projekt, falls noch kein Projekt existiert, mit dem Sie SmaSTra testen wollen. Klicken Sie in SmaSTra in der oberen Menüleiste auf "Tree" und wählen Sie die Option "Generate Java code...". Navigieren Sie zu ihrem Android Studio Projekt und wählen dort den Ordner "app" als Speicherort aus. Geben Sie den Namen ein, den die neue Java Klasse erhalten soll (in diesem Beispiel nennen wir sie "MovementSensor") und klicken Sie anschließend auf "Speichern". Wenn das Android Studio Projekt korrekt erkannt wurde, bittet ein Dialog um Erlaubnis, um den generierten Code automatisch in das gewählte Projekt zu integrieren. Bestätigen Sie und klicken im anschließenden Dialog auf "Ok".

Starten Sie Android Studio, falls nicht schon geschehen, und öffnen Sie das Projekt, in das gerade der SmaSTra Code exportiert wurde. Wählen Sie "Tools" in der oberen Menüleiste, dort auf Android und anschließend auf "Sync Project with Gradle Files". Sobald die Synchronisation abgeschlossen ist, kann die MovementSensor Klasse in ihrem eigenen Code verwendet werden.

Erstellen Sie dafür eine Instanz von MovementSensor und rufen Sie auf ihr die Methode start() auf. Ab jetzt kann der Wert des in SmaSTra gebauten MovementSensor einfach durch die Methode getData() abgefragt werden. Der Code in Figure ?? zeigt ein Beispiel, wie die Ausgabe des MovementSensor überprüft werden kann. Fügen Sie den Code in die onCreate Methode ihrer MainActivity mit ein und testen Sie das Programm auf einem angeschlossenen Smartphone oder Tablet. Die Ausgabe in der Konsole

sollte durchgehend "false" ausgeben, solange das Gerät nur wenig bewegt wird. Aber auch bei schnellen Bewegungen scheint die Ausgabe nicht immer auf "true" zu wechseln. Genauere Tests zeigen, dass der MovementSensor nur dann korrekt "true" ausgibt, wenn das Gerät schnell genug nach rechts, oben oder auf den Nutzer zu bewegt wird. Bewegungen nach links, unten und vom Nutzer weg, werden noch nicht richtig erkannt. Diesem Problem widmet sich der nächste Abschnitt.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final MovementSensor myMovementSensor = new MovementSensor(this);
    myMovementSensor.start();

    Timer timer = new Timer();
    timer.schedule(new TimerTask() {
        @Override
        public void run() {
            Log.d("SmaSTra Test", "Device is Moving: " + myMovementSensor.getData());
        }
    }, 1000, 1000);
}
```

Figure 2.6: Java Code zum Testen der in SMaSTra erstellten MovementSensor Klasse

Erstellen von Custom Elementen

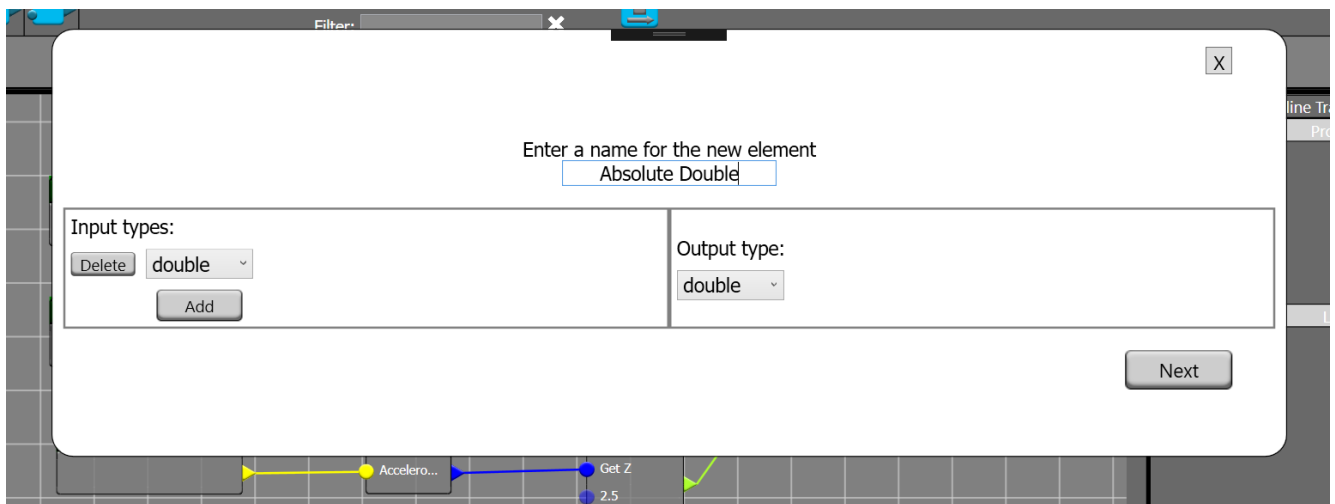


Figure 2.7: Erste Seite zum erstellen eines eigenen Elements. Hier werden der Name sowie Eingabe- und Ausgabetypen festgelegt

Ein Linear Accelerometer Sensor gibt für jede der drei Richtungsachsen X,Y und Z einen Wert aus. Bewegt sich das Gerät in positiver Richtung entlang einer Achse (z.b. nach rechts im Fall der X-Achse), wird auch ein positiver Wert ausgegeben. Eine Bewegung in negativer Richtung führt zu einem negativen Wert. Der bisher erstellte Java Code reagiert jedoch nur auf positive Werte, da negative Werte niemals den Schwellenwert von 2.5 überschreiten können. Eine mögliche Lösung wäre es, alle bisher verwendeten Elemente zu kopieren, anstatt Is Greater das

```

public static double AbsoluteDouble(double arg0) {
    if(arg0 < 0) arg0 = -arg0;
    return arg0;
}

```

Package name

Description


Figure 2.8: Zweite Seite zum erstellen eines eigenen Elements. Hier werden der auszuführende Code, Package-name und Beschreibung eingegeben

Element Is Smaller mit einem Wert von -2.5 zu verwenden und die beiden Teilbäume mit einem Bool 2 Or zu verbinden. Allerdings gibt es auch eine elegantere Lösung. Dafür wird ein Element benötigt, dass eine potenziell negative Ausgabe von Get X/Y/Z in einen positiven Wert umwandelt. Ein solches Element findet sich allerdings nicht in der Auswahlliste, daher muss es vom Nutzer selbst erstellt werden. Über "Edit" und anschließend "Create Custom Element" erscheint das Fenster zum erstellen eigener Elemente. Hier kann zunächst der Name des Elements, sowie seine Eingabe- und Ausgabetypen festgelegt werden (vgl. Figure ??). In diesem Beispiel wird das Element "Absolute Double" genannt. Als Eingabe- bzw. Ausgabetypen wird jeweils "double" verwendet. Mit einem Klick auf "Next" gelangt man zur nächsten Seite. Hier kann nun der Java/Android Code eingegeben werden, den das Element ausführen soll. Der Methodenkopf ist bereits vorgegeben. Der Eingabeparameter "arg0" soll in eine positive Zahl umgewandelt werden, falls er negativ ist. Der Code wäre demnach beispielsweise:

```

if(arg0 < 0) arg0 = -arg0;
return arg0;

```

Geben Sie anschließend noch einen beliebigen Packagenamen und eine Beschreibung des Elements an und Klicken Sie auf "Generate" (vgl. Figure ??). Dadurch wird eine neu Conversion erstellt, und der Auswahlliste hinzugefügt. Bisher werden allerdings vorinstallierte Elemente angezeigt. Mit einem Klick auf die Schaltfläche "Custom Elements"  werden auch Elemente angezeigt, die vom Nutzer selbst erstellt wurden. Das neue Element Absolute Double sollte nun auch bei den Conversions auftauchen. Schalten Sie je ein Absolute Double Element zwischen Get X/Y/Z und Is Greater. Sollte der Arbeitsbereich unübersichtlich werden, kann er jederzeit durch die Tastenkombination Strg+F geordnet werden (vgl. Figure ??). Anschließendes Generieren und Testen der Java Klasse zeigt, dass die Klasse genau dann "true" ausgibt, wenn das Gerät stark in beliebiger Richtung beschleunigt wird, was genau dem gewünschten Verhalten entspricht.

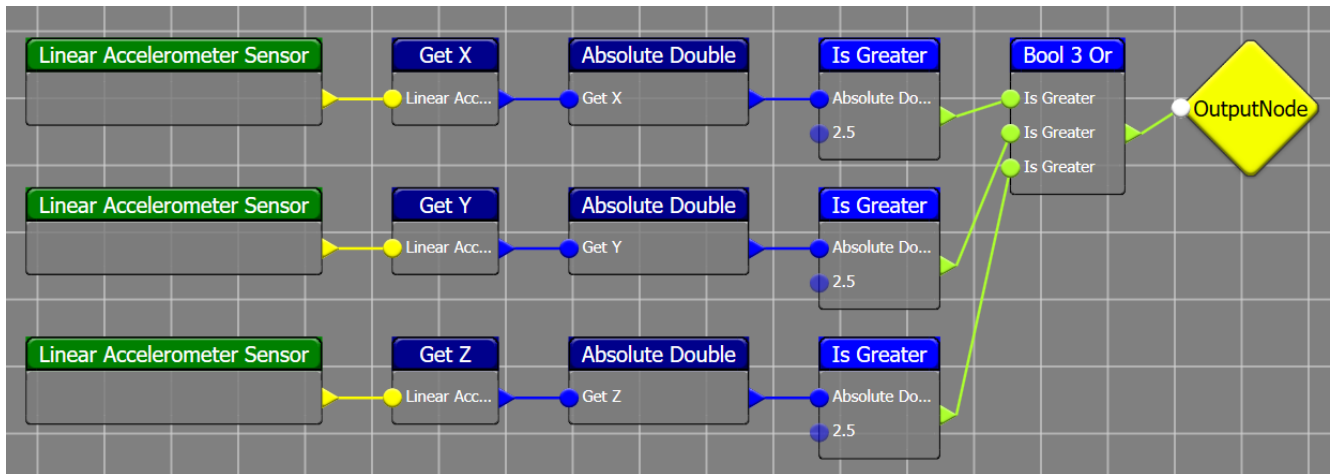


Figure 2.9: Arbeitsbereich nach dem Hinzufügen von Absolute Double und anschließendem Ordnen

Erstellen von Combined Elementen

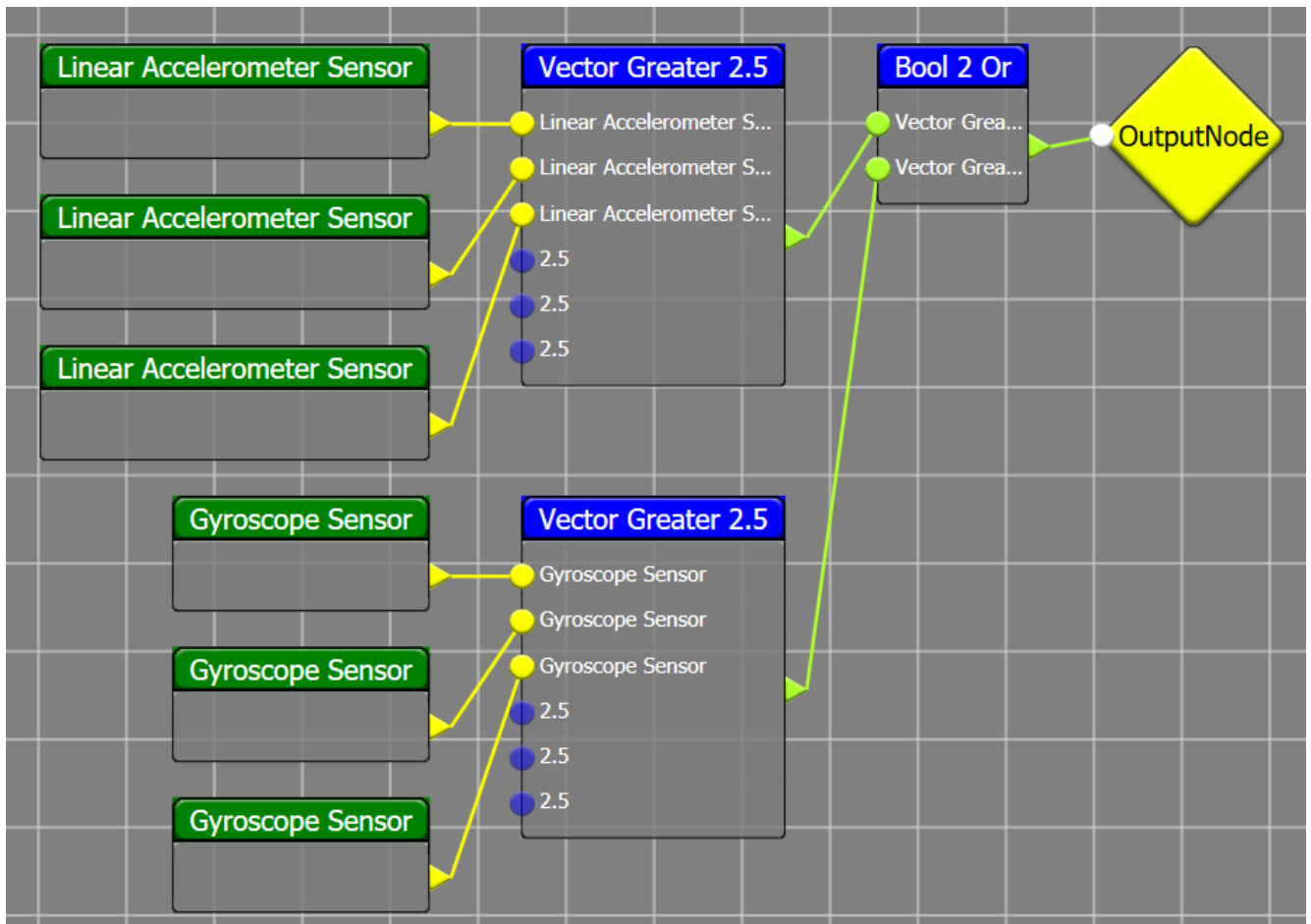



Figure 2.10: Arbeitsbereich nach Erstellen von Vector Greater 2.5 und Verbinden beider Zweige mit Output

Das obige Beispiel soll nun noch erweitert werden, um auch auf Rotation zu reagieren. Zusätzlich zum bisherigen Verhalten soll die generierte Java Klasse auch dann "true" ausgeben, wenn das Gerät mit mehr als 2.5 rad/s um eine beliebige Achse rotiert. Diese Aufgabe ist der vorangegangenen sehr ähnlich und auch die benötigten Elemente sind fast identisch. Ein Lösungsansatz wäre es, einfach jedes einzelne Element zu kopieren und damit einen zweiten Baum zu erstellen, bei dem die drei Linear Accelerometer




Sensoren durch Gyroscope Sensoren ersetzt werden. Diese Vorgehensweise ist zwar prinzipiell korrekt, führt aber zu einer hohen Anzahl redundanter Elemente.




Doch anstatt jedes Element einzeln zu kopieren, ist es möglich eine Gruppe von verbundenen Elementen zu einem einzelnen Element zusammenzufassen. Ziehen Sie mit der linken Maustaste einen Rahmen um alle Elemente, ausgenommen dem Output und den drei Linear Accelerometer Sensoren. Klicken Sie dann mit der rechten Maustaste auf eines der markierten Elemente und wählen Sie im Kontext Menü "Merge current selection". Geben Sie in dem erscheinenden Fenster einen Namen für das neue Element an (in diesem Beispiel wird es "Vector Greater 2.5" genannt) und bestätigen Sie mit "OK". Aus den zuvor markierten Elementen wurde nun ein einzelnes Element generiert. Elemente die auf diese Weise erstellt werden, befinden sich in der Auswahlleiste in einer eigenen Kategorie. Klicken Sie auf die Schaltfläche "Combined Elements"  um diese Elemente anzeigen zu lassen. Ziehen Sie ein zweites Vector Greater 2.5 Element zusammen mit drei Gyroscope Sensoren in den Arbeitsbereich. Verbinden Sie die Gyroscope Sensoren mit dem zweiten Vector Greater 2.5 analog zu den Linear Accelerometer Sensoren. Abschließend wird noch ein "Bool 2 Or" benötigt. Verbinden Sie dieses mit den jeweiligen Ausgabesockets der beiden Vector Greater 2.5 und dem Output (vgl. Figure ??). Damit ist schon die nächste Java Klasse bereit, um erstellt und getestet zu werden. Wie gewünscht sollte nun auch bei einer entsprechend hohen Rotationsgeschwindigkeit des Geräts "true" ausgegeben werden.

2.3 Einzelanleitung zu jeder Funktion

2.3.1

2.3.2 Auswahlleiste

Die Auswahlleiste beinhaltet sämtliche Elemente, die im aktuellen Workspace gespeichert sind. Im Regelfall sind dies alle vorinstallierten Elemente, sowie den vom Nutzer erstellten eigenen Elementen und den kombinierten Elementen. Um Elemente besser einordnen zu können, werden sie jeweils nach zwei Kriterien unterschieden. Das erste ist die Anzahl ihrer Eingabesockets. Elemente ohne ein Eingabesocket werden als "Datasources" bezeichnet. Elemente mit genau einem Eingabesocket sind "Conversions" und mit mehr als einem Eingabesocket sind "Transformations". Mit den jeweiligen Schaltflächen oberhalb der Auswahlleiste lassen sich die Datasources , Conversions  und Transformations  in der Auswahlleiste ein und ausblenden.

Das zweite Kriterium ist die Art wie das jeweilige Element erstellt wurde. Vorinstallierte Elemente werden als "Default Elements" bezeichnet, vom Nutzer erstellte Elemente sind "Custom Elements" und zusammengefasste Elemente werden als "Combined Elements" bezeichnet. Auch hier können Elemente einer bestimmten Kategorie mit den Schaltflächen Default Elements , Custom Elements  und Combined Elements  ein oder ausgeblendet werden.

Zusätzlich zu den beiden Kriterien, kann auch nach den Namen der Elemente gefiltert werden. Sobald eine Zeichenfolge in das "Filter" Textfeld eingegeben wird, werden nur noch Elemente angezeigt, die diese Zeichenfolge in ihrem Namen beinhalten. Der Filter kann entweder durch Löschen aller Zeichen im Textfeld oder durch Drücken der X-Taste neben dem Textfeld zurückgesetzt werden.

Von Nutzern erstellte Elemente (also aus den Kategorien Combined Elements und Custom Elements) können aus der Auswahlleiste gelöscht werden. Klicken Sie dazu mit der rechten Maustaste auf das zu entfernende Element und wählen Sie die Option "Delete". Ein auf diese Weise entferntes Element wird der node.blacklist Datei des aktuellen Workspace hinzugefügt, wodurch es nicht mehr in der Auswahlleiste erscheint. Sie können Einträge aus der Datei löschen, um das Element der Auswahlleiste wieder hinzuzufügen.

Um ein Element entgültig zu löschen, muss der Ordner "created/[Name des Elements]" im aktuellen Workspace gelöscht werden. Seien Sie dabei jedoch sicher, dass das Element nicht weiter benötigt wird und auch in keinen zusammengesetzten Elementen verwendet wird.

2.3.3 Interaktionen mit Elementen

Elemente in den Arbeitsbereich ziehen

Um ein Element aus der Auswahlleiste in den Arbeitsbereich zu ziehen, bewegen Sie den Mauszeiger über das Element, Klicken und halten Sie die linke Maustaste gedrückt und bewegen Sie den Mauszeiger an die Stelle im Arbeitsbereich, an der das Element erscheinen soll. Lassen Sie die linke Maustaste nun los um das Element zu platzieren.

Verbinden von Elementen

Jedes Element (abgesehen des Outputs) besitzt genau einen Ausgabesocket und eine Variable Anzahl an Eingabesockets. Jeder Socket hat einen festgelegten Datentyp, der die Farbe des Sockets bestimmt. Der Text neben unverbundenen Eingabesockets zeigt die Bezeichnung des Datentyps an. Ein Ausgabesocket kann nur mit einem Eingabesocket verbunden werden und umgekehrt. Zudem müssen die Datentypen der beiden Sockets gleich sein (also die selbe Farbe haben). Die einzige Ausnahme bildet dabei wieder der Output, dessen Eingabesocket mit jedem beliebigen Ausgabesocket verbunden werden kann.

Um im Arbeitsbereich zwei Elemente bzw. ihre Sockets zu verbinden, bewegen Sie den Mauszeiger über eines der beiden Sockets, dass Sie verbinden möchten. Klicken und halten Sie dann die linke Maustaste gedrückt und bewegen Sie den Mauszeiger über den anderen Socket. Während der Bewegung werden alle kompatiblen Sockets grün umrandet, während alle anderen Sockets rot umrandet werden. Zudem weist eine blaue Linie vom gehaltenen Socket zum Mauszeiger darauf hin, dass gerade eine Verbindung durchgeführt wird. Befindet sich der Mauszeiger über dem anderen Socket, lassen Sie die linke Maustaste los und die Verbindung wird erstellt, sofern die beiden Sockets kompatibel sind. Die Verbindungslinie erscheint nun in der selben Farbe wie die beiden Sockets. Der Text neben dem verbundenen Eingabesocket zeigt nun den Namen des Elements an, mit dem es verbunden wurde.

Wird eine Verbindung von bzw. zu einem Socket aufgebaut, das bereits verbunden ist, wird die alte Verbindung automatisch gelöst.

Anstatt ein Eingabesocket mit einem Ausgabesocket zu verbinden, kann auch ein fester Wert als Eingabeparameter festgelegt werden. Führen Sie dafür einen Doppelklick auf den Eingabesocket aus, für den ein Wert eingetragen werden soll. Dabei wird das ausgewählte Element in den Properties angezeigt und das Textfeld des entsprechenden Eingabesockets markiert (siehe auch ??). Nun können Sie einen Wert eintragen. Dieser Wert ersetzt dabei die Anzeige des Datentyps neben dem Eingabesocket. Zusätzlich wird der Eingabesocket transparent. Eine eventuell bestehende Verbindung an diesem Socket wird dabei automatisch gelöst.

Bewegen von Elementen

Jedes Element kann im Arbeitsbereich frei bewegt werden. Bewegen Sie dazu den Mauszeiger über das Element und halten Sie die linke Maustaste gedrückt. Das Element folgt nun jeder Bewegung des Mauszeigers, so lange Sie die Maustaste gedrückt halten. Sollten mehrere Elemente ausgewählt sein, werden alle gleichzeitig bewegt.

Auswählen von Elementen

Ein Element wird ausgewählt, indem es einmal mit der linken Maustaste angeklickt wird. War bereits ein anderes Element ausgewählt, verliert dieses seine Auswahlstatus. Sollen mehrere Elemente ausgewählt werden, gibt es dafür mehrere Möglichkeiten. Ein Element kann einer bereits bestehenden Auswahl hinzugefügt werden, indem man die Strg-Taste gedrückt hält während man das Element

anklickt. Alternativ kann das Element auch mit der rechten Maustaste angeklickt und die Option "Add to selection" gewählt werden. Wird ein bereits ausgewähltes Element bei gedrückter Strg-Taste angeklickt, wird es aus der Auswahl entfernt.

Man kann einen Rahmen um beieinander liegende Elemente ziehen, um dies auszuwählen. Klicken Sie dafür mit der linken Maustaste auf eine freie Stelle im Arbeitsbereich und halten Sie die Taste gedrückt. Wenn Sie die Maustaste nun bewegen, wird dabei ein Rahmen gezogen. Sobald Sie die Maustaste wieder loslassen, werden alle Elemente markiert, deren jeweiliger Mittelpunkt innerhalb des Rahmens liegt. Schließlich kann eine Gruppe von Elementen, die aktuell verbunden sind ausgewählt werden, indem ein Doppelklick auf ein beliebiges Element der Gruppe ausgeführt wird. Alternativ kann ein beliebiges Element der Gruppe auch mit der rechten Maustaste angeklickt und die Option "Select all connected" gewählt werden.

Entfernen von Elementen

Ausgewählte Elemente können mit der Entf-Taste wieder aus dem Arbeitsbereich entfernt werden. Alternativ kann mit einem Rechtsklick auf ein ausgewähltes Element im Kontextmenü die Option "Delete" ausgewählt werden. Verbindungen zu entfernten Elementen werden automatisch getrennt. Der Output kann nicht entfernt werden.

Kopieren von Elementen

Ausgewählte Elemente können mit Strg+V kopiert werden. Jede Kopie wird dabei automatisch der Auswahl hinzugefügt, während die Originale aus der Auswahl entfernt werden. Bestehende Verbindungen werden nicht mit kopiert. Per Hand eingetragene Werte der Eingabesockets bleiben jedoch bei den Kopien erhalten. Der Output kann nicht kopiert werden.

Zusammenfassen von Elementen

Eine Gruppe von verbundenen Elementen können zu einem einzelnen Element zusammengefasst werden. Wählen Sie alle Elemente aus, die zusammengefasst werden sollen und klicken Sie dann mit der rechten Maustaste auf eines der Elemente. Wählen Sie im Kontextmenü die Option "Merge current selection". Bestätigen Sie den Dialog und geben Sie einen Namen für das neue Element an. Alle zuvor markierten Elemente sind nun zu einem neuen Element zusammengefasst worden. Die Eingabesockets des neuen Elements entsprechen allen Eingabesockets, die bei den markierten Elementen nicht verbunden waren. Per Hand eingetragene Werte bleiben dabei erhalten. Das neue Element wird automatisch in der Auswahlleiste den Combined Elements hinzugefügt. Der Output kann nicht mit zusammengefasst werden.

Ein zusammengefasstes Element kann im Arbeitsbereich auch wieder in seine einzelnen Elemente zerlegt werden. Klicken Sie dafür mit der rechten Maustaste auf das Element und wählen Sie die Option "unmerge current selection". Der Eintrag in der Auswahlleiste bleibt dabei weiterhin bestehen.

2.3.4 Properties

Die Properties befinden sich rechts neben dem Arbeitsbereich. Hier werden Details des aktuell ausgewählten Elements angezeigt:

- **Name:** Die Bezeichnung des Elements. Der Name kann im Textfeld beliebig vergeben werden. Der Standardwert entspricht der Klasse des Elements.
- **Class:** Die Klasse des Elements. Dieser Wert ist festgelegt und dient dazu den Typ eines Elements zu bestimmen, wenn dessen Name geändert wurde.
- **Input:** Hier werden die Typen und Werte der Eingabesockets angezeigt. Der Typ ist festgelegt. Der Wert kann über das Textfeld eingetragen werden.

-
- Output: Gibt des Typ des Ausgabesockets an.
 - X/Y: Die Position des Elements im Arbeitsbereich. Die Werte können über das Textfeld geändert werden, um die Position genau festzulegen.
 - Description: Die Beschreibung des Elements. Hier wird dem Nutzer mitgeteilt, welche Funktion das Element erfüllt und welche Bedeutung die jeweiligen Eingabe- und Ausgabesockets haben. Elemente, die vom Nutzer selbst erstellt werden, sollten immer mit einer ausführlichen Beschreibung versehen werden, um anderen Nutzern ihre Bedeutung und Verwendung verständlich zu machen.

Der Propertiesbereich kann über die Schaltfläche "Properties" ein- und ausgeblendet werden.

2.3.5 Library

Die Library befindet sich am rechten Rand des Fensters, unterhalb der Properties. Hier können häufig verwendete Elemente gelagert werden, um jederzeit schnellen Zugriff auf diese Elemente zu gewährleisten. Elemente aus der Auswahlleiste können direkt in die Library gezogen werden. Um der Library Elemente aus dem Arbeitsbereich hinzuzufügen, klicken Sie mit der rechten Maustaste auf das Element und wählen Sie die Option "Add to Library". Elemente werden aus der Library analog zur Auswahlleiste in den Arbeitsbereich gezogen. Um ein Element aus der Library zu entfernen, klicken Sie mit der rechten Maustaste auf das Element in der Library und wählen Sie die Option "Delete". Das Element bleibt in der Auswahlleiste weiterhin bestehen. Die Library kann mit einem Klick auf die "Library" Schaltfläche ein- und ausgeblendet werden.

2.3.6 Erstellen Eigener Elemente

Es besteht die Möglichkeit eigene Elemente zu erstellen, indem das Verhalten des Elements mit Java/Android Code festgelegt wird. Wählen Sie dafür in der obersten Menüleiste "Edit" und dort "Create Custom Element". Dadurch wird die erste Seite des Fensters zum Erstellen von eigenen Elementen angezeigt. Hier kann der Name des neuen Elementtyps angegeben werden. Vergeben Sie einen möglichst beschreibenden aber dennoch kurzen Namen. Der Name darf zudem nicht identisch zu einem bereits bestehenden Elementtyp sein.

Auf dieser Seite werden auch die Eingabe- und Ausgabesockets des Elements festgelegt. Wählen Sie aus der Combobox den gewünschten Datentyp des Sockets. Sollte kein passender Datentyp dabei sein, können Sie "[Add new Datatype]" auswählen und in das erscheinende Textfeld den Namen eines neuen Datentyps eingeben. Sie können weitere Eingabesockets über die Schaltfläche "Add" hinzufügen, bzw. bestehende Eingabesockets über "Delete" entfernen. Klicken Sie anschließend auf "Next" um zur zweiten Seite zu gelangen.

Hier kann der eigentliche Java Code eingegeben werden, der vom Element ausgeführt werden soll. Der Methodenkopf ist bereits durch die ausgewählten Eingabesockets festgelegt. Die Eingabeparameter werden mit `arg0`, `arg1`, usw. bezeichnet. Es werden keinerlei semantischen oder syntaktischen Überprüfungen bezüglich des eingegebenen Codes vorgenommen. Beachten Sie auch, dass die öffnende und schließende Klammer der Methode bereits vorgegeben sind.

Nach der Eingabe des Codes müssen noch der Name des Package und abschließend eine Beschreibung des neuen Elements eingetragen werden. Beschreiben Sie das Verhalten des Elements, sowie die Bedeutung aller Eingabe- und Ausgabesockets ausführlich, um anderen Nutzern den Zweck und den korrekten Umgang mit ihrem neuen Element zu erläutern. Klicken Sie dann auf "Generate" um das Element zu erstellen. Es wird in der Auswahlleiste unter der Kategorie Custom Element auftauchen.

2.3.7 Generieren und Exportieren des Java Codes

Um den in SmaSTra gebauten Sensor in einem eigenen Android Projekt verwenden zu können, muss der Code erst aus den Elementen generiert und anschließend in das Projekt exportiert werden. Stellen Sie sicher, dass mindestens ein Element im Arbeitsbereich mit dem Output verbunden ist. Zum Generieren des Codes, klicken Sie in der oberen Menüleiste auf "Tree" und wählen dort die Option "Generate Java code...". Geben Sie einen Namen für die zu erstellende Java Klasse ein. Der Speicherort sowie der weitere Verlauf hängt von der Entwicklungsumgebung ab, mit dem Sie ihr Projekt entwickeln:

Exportieren in Android Studio

SmaSTra ist in der Lage Android Studio Projekte zu erkennen und die generierten Dateien und Ordner automatisch in die entsprechenden Stellen des Projekts zu integrieren. Wählen Sie als Speicherort den "app" Ordner in Ihrem Android Studio Projekt. Wenn SmaSTra das Projekt erkennt, bittet ein Dialog um Erlaubnis, die erstellten Dateien und Ordner in das Projekt zu exportieren. Bestätigen Sie diesen Dialog. Ein weiterer Dialog informiert Sie sobald der Export abgeschlossen wurde. Starten Sie in Android Studio anschließend eine Gradle Synchronisation, über "Tools" -> "Android" -> "Sync Project with Gradle Files". Sobald diese Abgeschlossen ist, kann die erstellte Klasse verwendet werden.

Exportieren in Eclipse

Im Vergleich zu Android Studio ist das Exportieren in Eclipse aufwändiger, da Eclipse zum aktuellen Zeitpunkt keine Android Archive Librarys (.aar Dateien) unterstützt. Wählen Sie einen beliebigen temporären Ordner als Speicherort für den generierten Code.

Zunächst muss ein Android Library Projekt für die SmaSTraBase.aar Library erstellt werden. Falls Sie eine solche Library schon aus vorangegangenen Projekten besitzen, können Sie diese wieder verwenden und daher diesen Teil überspringen. Bei einer .aar Datei handelt es sich um ein Archiv. Entpacken Sie dieses in einen beliebigen Ordner (hier wird er "SmaSTraBase" genannt). Erstellen Sie außerdem einen neuen Ordner (hier "SmaSTra Lib"), der als Projektordner für das SmaSTra Library Projekt dienen soll. Kopieren Sie nun aus dem SmaSTraBase Ordner die "AndroidManifest.xml" Datei, sowie die Ordner "assets", "libs" und "res" in den SmaSTra Lib Ordner. Fügen Sie dem libs Ordner außerdem die classes.jar Datei aus dem SmaSTraBase Ordner hinzu. Der SmaSTra Libs Ordner soll nun als Library Projekt in Eclipse importiert werden. Öffnen Sie dafür Eclipse und wählen Sie "File" -> "Import". Im erscheinenden Dialog wählen Sie in der Kategorie "Android" die Option "Existing Android Code Into Workspace" und klicken Sie auf "Next". Über die Schaltfläche "Browse..." können Sie den Ordner SmaSTra Lib auswählen. Aktivieren Sie den Hacken bei "copy projects into workspace" und klicken Sie auf "finish". Im Package Explorer sollte nun das Projekt "SmaSTra Lib" auftauchen. Klicken Sie mit der rechten Maustaste auf das Projekt und wählen Sie Properties. Im auftauchenden Dialog klicken Sie auf "Android" und setzen Sie den Hacken bei "Is Library". Das SmaSTra Lib Projekt ist damit ein Library Projekt und kann für alle SmaSTra Projekte wiederverwendet werden.

Um die SmaSTra Library mit ihrem Eclipse Projekt zu verbinden, klicken Sie mit der rechten Maustaste auf ihr Eclipse Projekt und wählen "Properties". Unter der Kategorie "Android" klicken Sie im Bereich Library auf "Add", wählen Sie ihr SmaSTra Library Projekt aus ("SmaSTra Lib" falls Sie sich am obigen Absatz orientiert haben) und bestätigen Sie mit "ok".

Nun müssen nur noch die Quellcode Dateien an die richtige Stelle kopiert werden. Öffnen Sie den Speicherort, in dem Sie die SmaSTra Dateien generiert haben. Kopieren Sie die Ordner "Test", "de" und "custom" (falls vorhanden) in den "src" Ordner ihres Eclipse Projekts. Klicken Sie dann in Eclipse mit der rechten Maustaste auf ihr Projekt und wählen Sie "Refresh". Eclipse sollte die hinzugefügten Ordner im Package Explorer nun anzeigen. Damit ist die erstellte Klasse bereit zur Verwendung.

2.3.8 Verwendung der generierten Java Klasse

Nachdem der Generierte SmaSTra Code in ein Projekt integriert wurde (siehe Abschnitt ??), kann er in diesem Projekt verwendet werden. Im folgenden wird davon ausgegangen, dass die generierte Klasse "SmastraSensor" genannt wurde. Erstellen Sie in ihrem Code eine neue Instanz vom Typ SmaSTraSensor. der Konstruktor erwartet dabei den aktuellen Context.


```
SmastraSensor mySensor = new SmastraSensor(getApplicationContext());
```

Auf diese Instanz können anschließend drei Methoden aufgerufen werden:

- start(): Diese Methode initialisiert die Hardware Sensoren des Geräts. Rufen Sie diese Methode auf, sobald Sie damit beginnen wollen, den Ausgabewert ihres Sensors abzufragen.
- getData(): Diese Methode liefert den Ausgabewert des Sensors zurück. Beachten Sie dass zuvor start() aufgerufen worden sein muss.
- stop(): Diese Methode gibt die Hardware Sensoren frei. Rufen Sie diese Methode auf, wenn Sie den Sensor nicht weiter benötigen.

2.3.9 Online Repository

Interaktion mit dem Online Menü

Um selbsterstellte Elemente mit anderen Nutzern zu teilen, besteht die Möglichkeit Elemente auf einen Webserver hoch und von diesem wieder herunterzuladen. Durch die Schaltfläche "Online Menu"  öffnet sich das Fenster zum Datenaustausch mit dem Server. Die selbe Funktionalität bietet auch der Bereich "Online" entlang der rechten Fensterseite über den Properties. Im oberen linken Abschnitt befindet sich eine Liste der verfügbaren Elemente des Servers. Wählen Sie ein Element aus, um dessen Details rechts neben der Liste anzeigen zu lassen. Über die Schaltfläche "Download" lässt sich das gewählte Element herunterladen und ist anschließend in der Auswahlleiste zu finden. Um ein Element auf den Server hochzuladen, bewegen Sie den Mauszeiger in der Auswahlleiste über das entsprechende Element. Klicken und halten Sie die linke Maustaste und bewegen Sie den Mauszeiger über den grauen "Upload Drop Zone" Bereich im Online Menü. Lassen Sie anschließend die linke Maustaste los, um das Element auf den Server zu laden. Die Liste der Server Elemente wird automatisch aktualisiert und beinhaltet nun auch das hochgeladene Element.

Einrichten des Servers

Um das Online Menü nutzen zu können, muss zunächst ein entsprechender Server eingerichtet und dieser mit dem aktuellen SmaSTra Workspace verbunden werden. Zur Verwendung mit einem Tomcat 8 oder Tomcat 9 Server, befindet sich bereits ein passendes Servlet (.war Datei) im Ordner "SmaSTraWeb-Server".

Um das Servlet auf dem Server zu installieren, starten Sie den Tomcat Server und öffnen Sie den Server über einen Webbrowser. In der sich öffnenden Webseite klicken Sie auf "Tomcat Manager" und anschließend auf "WAR file to deploy". Wählen Sie nun die .war Datei aus dem SmaSTraWebServer Ordner und laden Sie die Datei über "Deploy" auf den Server hoch.

Nun muss noch der SmaSTra Workspace für die Verwendung mit dem Server konfiguriert werden. Stellen Sie sicher, dass SmaSTra aktuell nicht geöffnet ist. Navigieren Sie mit einem Datei-Explorer zum verwendeten SmaSTra Workspace. Öffnen Sie die Datei "config.prop" in einem Texteditor. Tragen Sie die Adresse oder IP des Servers unter "onlineServiceHost" ein (z.B.: onlineServiceHost=http://localhost). Tragen Sie den Port des Servers unter "onlineServicePort" ein. Falls Sie das Servlet aus dem Ordner

SmaSTraWebServer verwenden, ist der Standardport 80. Tragen Sie das Prefix des Servlets unter "onlineServicePrefix" ein. Falls Sie das Servlet aus dem Ordner SmaSTraWebServer verwenden, ist der Standardwert "SmaSTraWebServer".

Damit sind der Server und der SmaSTra Workspace verbunden und das Online Menü wird mit dem nächsten Start von SmaSTra funktionsfähig sein.

2.3.10 Workspaces

Um den Wechsel zwischen mehreren Nutzern und unterschiedlichen Projekten zu unterstützen, kann der aktuelle Workspace geändert werden. Klicken Sie dafür in der obersten Menüleiste auf "File" und dort die Option "Switch Workspace". Wählen Sie im auftauchenden Dialog den Ordner, der als Workspace verwendet werden soll.

Im Workspace werden unter anderem folgende Informationen gespeichert:

- Der Ordner "created" beinhaltet alle Elemente, die von Nutzern erstellt wurden.
- Der Ordner datatypes beinhaltet alle Datentypen, die keine primitiven Typen sind.
- Die Datei "config.prop" enthält die Informationen bezüglich des Servers zum Austausch von Elementen. Genauere Informationen hierzu finden Sie im Abschnitt ??
- In der Datei "current.work" wird der Arbeitsbereich beim Schließen von SmaSTra gespeichert, so dass beim nächsten Start die Arbeit an der selben Stelle weiter geführt werden kann.
- Die Datei "node.blacklist" listet alle Elemente auf die aus der Auswahlleiste gelöscht wurden. Löschen Sie Einträge aus dieser Liste, um Elemente wieder der Auswahlleiste hinzuzufügen.

2.3.11 Shortcut Überblick

Kamera bewegen

Alt + Mausrad	Zoomen
Alt + O	Zum Output springen
Alt + Maus ziehen	Arbeitsbereich ziehen

Aktionen

STRG + M	Ausgewählte Elemente zusammenfassen
STRG + U	Ausgewähltes Element aufteilen
Alt + W	Workspace wechseln
STRG + S	Speichern
STRG + L	Laden
Alt + F4	Beenden
STRG + G	Java Code Generieren
Entf	Ausgewählte Objekte Löschen
STRG + F	Elemente Ordnen

Elemente auswählen

STRG + Linksklick	Element zur Auswahl hinzufügen
Doppelklick	Verbundene Elemente auswählen

3 Anhang

3.1 Evaluation

6 Antworten

Zusammenfassung

Vor der Evaluation

Hast du Ahnung vom Programmieren?



Ja 2 33.3 %
Nein 4 66.7 %

Hast du Ahnung von Android Programmierung?



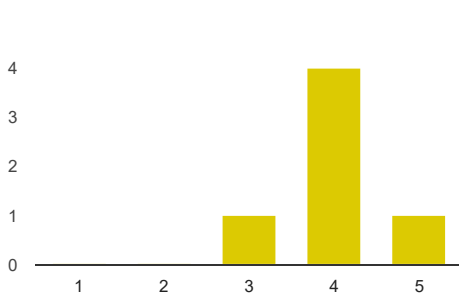
Ja 1 16.7 %
Nein 5 83.3 %

Falls Ja, ist es schneller mit SmaSTra auf die Sensordaten zu zugreifen?



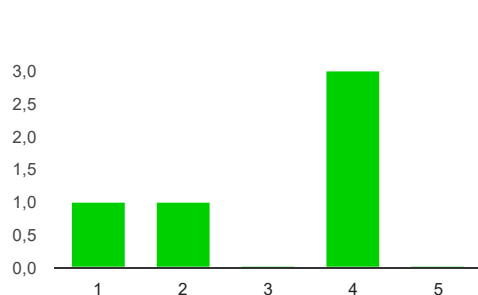
Ja 1 100 %
Nein 0 0 %

Wie fandest du die Menu Führung?



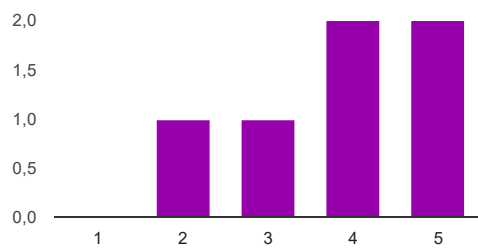
Schlecht: 1 0 0 %
2 0 0 %
3 1 16.7 %
4 4 66.7 %
Gut: 5 1 16.7 %

Wie fandest du das Auswahl Menu?



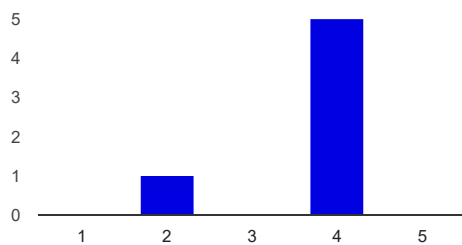
Schlecht: 1 1 20 %
2 1 20 %
3 0 0 %
4 3 60 %
Gut: 5 0 0 %

Fandest du die Benutzeroberfläche übersichtlich?



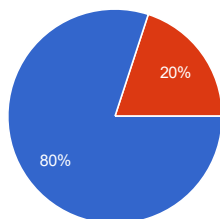
Schlecht: 1	0	0 %
2	1	16.7 %
3	1	16.7 %
4	2	33.3 %
Gut: 5	2	33.3 %

Fandest du die Benutzeroberfläche angenehmen(Farbe, Form etc)?



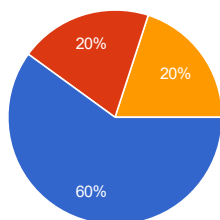
Schlecht: 1	0	0 %
2	1	16.7 %
3	0	0 %
4	5	83.3 %
Gut: 5	0	0 %

Waren die Funktionen von den Symbolen verständlich?



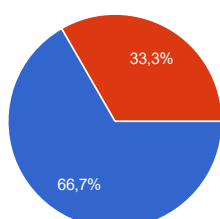
Ja	4	80 %
Nein	1	20 %
Sonstige	0	0 %

War das Hinzufügen von neuen Elementen auf der Arbeitsfläche intuitiv?



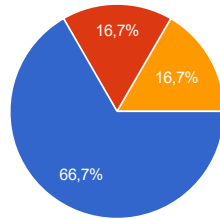
Ja	3	60 %
Nein	1	20 %
Sonstige	1	20 %

War gut zu erkennen, welche Elemente markiert sind?



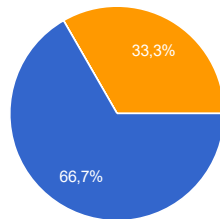
Ja	4	66.7 %
Nein	2	33.3 %
Sonstige	0	0 %

Waren die unterschiedliche Farben zwischen Input und Output hilfreich?



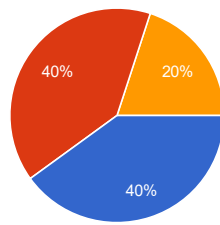
Ja	4	66.7 %
Nein	1	16.7 %
Sonstige	1	16.7 %

War die Merge-Funktion hilfreich?



Ja	4	66.7 %
Nein	0	0 %
Sonstige	2	33.3 %

Waren die Informationen von Properties hilfreich?



Ja	2	40 %
Nein	2	40 %
Sonstige	1	20 %

Waren Features nicht offensichtlich genug? Wenn ja, welche?

Nein

Was fehlte dir an Features?

Elemente groesser machen (z.B. weil Text nicht mehr ganz zu lesen ist). Text veraendern (z.B. kann man 'Accelerometer Sensor' nicht aendern, ich wollte naemlich schreiben Acc X, auch 2.5 koennte ich nicht mehr aendern).

Sonstige Anregungen?

I liked the program in general, nodes connecting, dragging and dropping etc were retty intuitive. I know very little about programming but from the design part I would definetely work a bit more. The interface design is a tiny bit boring and tools panel can be confusing sometimes. All the elements from the upper 3 buttons are in the same row. I would put them in different tabs.

Das Programm ist eigentlich schnell gut verstaendlich. Es gab keine Bugs. Ich fand es gut, dass es uebersichtlich und nicht ueberlagert ist.

Da ich bei Aufgabe 2 erst nicht genau gelesen hatte, musste ich einige Schritte per Undo rückgängig machen. Anschließend musste ich nach dem mergen anders benennen, da die erste (rückgängig gemachte) Benennung noch vorhanden/gespeichert war.

Anzahl der täglichen Antworten

2,0



