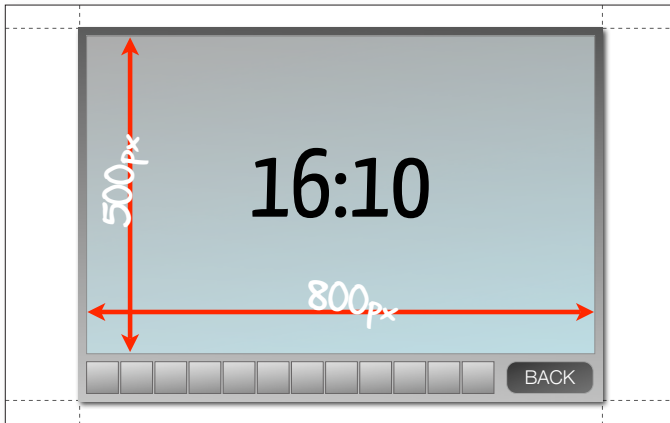
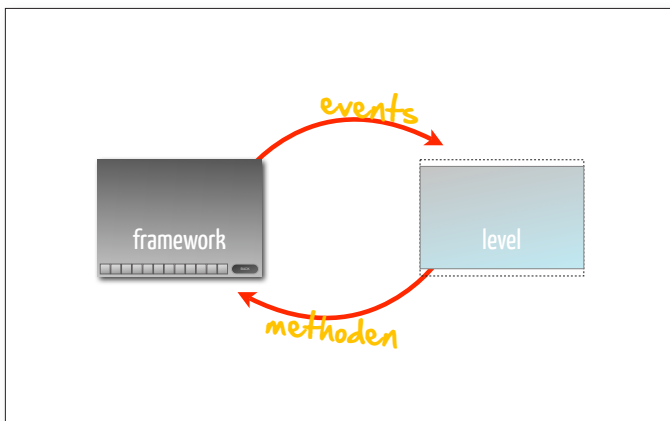


mp2id: framework



Erstellen Sie ihr Level im Format 16:10 mit den Abmessungen 800x500 Pixel. Rucksack und Backbutton werden vom Framework bereitgestellt



Die Kommunikation zwischen Framework und Level erfolgt über Events, während die Kommunikation in die andere Richtung im allgemeinen über Aufrufe bereitgestellter Methoden abgewickelt wird.

Das Framework stellt den optischen und funktionalen Rahmen für das Level zur Verfügung. Es lädt das 800x500 Pixel große Level und stellt ein Objekt zur Kommunikation bereit. Das Level kann durch entsprechende Methodenaufrufe mit dem Framework interagieren, das Framework wiederum informiert das Level über relevante Ereignisse durch das Auslösen von Events.

Template-Projekt verwenden (Flex)

Das Template-Projekt ist mit dem Flex-Builder 3 erzeugt worden, kann jedoch auch mit Flash Builder 4 verwendet werden. Zum Importieren des Projektes (File-Import-Flash/Flex Project) muss die ZIP-Datei ausgewählt werden. Beim Flash Builder 4 erscheint dann eine Meldung, dass eine früheres SDK verwendet wurde. Wählen Sie hier als neu zu verwendendes SDK unbedingt Flex 3.5 aus. Im Flex Builder 3 wird standardmäßig das korrekte SDK 3.2 verwendet.

Das Template-Projekt umfasst neben den üblichen Ordnern (src, libs, ...) auch noch fw (darin sind alle vom Framework zur Verfügung gestellten Dateien enthalten) und res (dieser Ordner kann für Dateien, die zur Laufzeit geladen werden, verwendet werden). Die Dateien in den Ordner fw und res werden beim Kompilieren in das build-Verzeichnis inkludiert. Damit beim Ausführen der Applikation das Framework geladen wird, wurde das html-Template entsprechend verändert (dort wird nun die Datei fw.swf geladen). Das Framework ist so konfiguriert, dass es seinerseits die Datei Level.swf als Level lädt. Daher muss Ihre mxmL-Applikation (sprich das Level) auch Level heißen.

Template-Projekt verwenden (Flash)

Sollten Sie in Flash entwickeln, nutzen Sie den zur Verfügung gestellten Loader. Hierbei gibt es zwei Versionen. Die eine Version funktioniert nur mit Flash CS4 und höher (FW-Loader fla). Diese lädt das im selben Ordner befindliche Framework (fw.swf) und dieses lädt wiederum das von Ihnen erstellte Level (Level.swf). Exportieren Sie immer zuerst die Level.swf und wechseln Sie dann zum Loader und exportieren bzw. starten Sie dann diesen. Trace-Ausgaben werden entsprechend ausgegeben, das Setzen von Haltepunkten ist jedoch nicht möglich.

Wenn sie Flash CS3 haben, verwenden Sie den anderen mitgelieferten Loader (FW-Loader-cs3 fla), dieser lädt eine speziell für Flash CS3 exportierte Version des Frameworks. Wenn Sie ihr Level trotzdem mit dem Original Framework betrachten wollen, nutzen Sie die mitgelieferte HTML-Seite (FW-Loader.html).

Allgemeines zum Aufbau des Levels

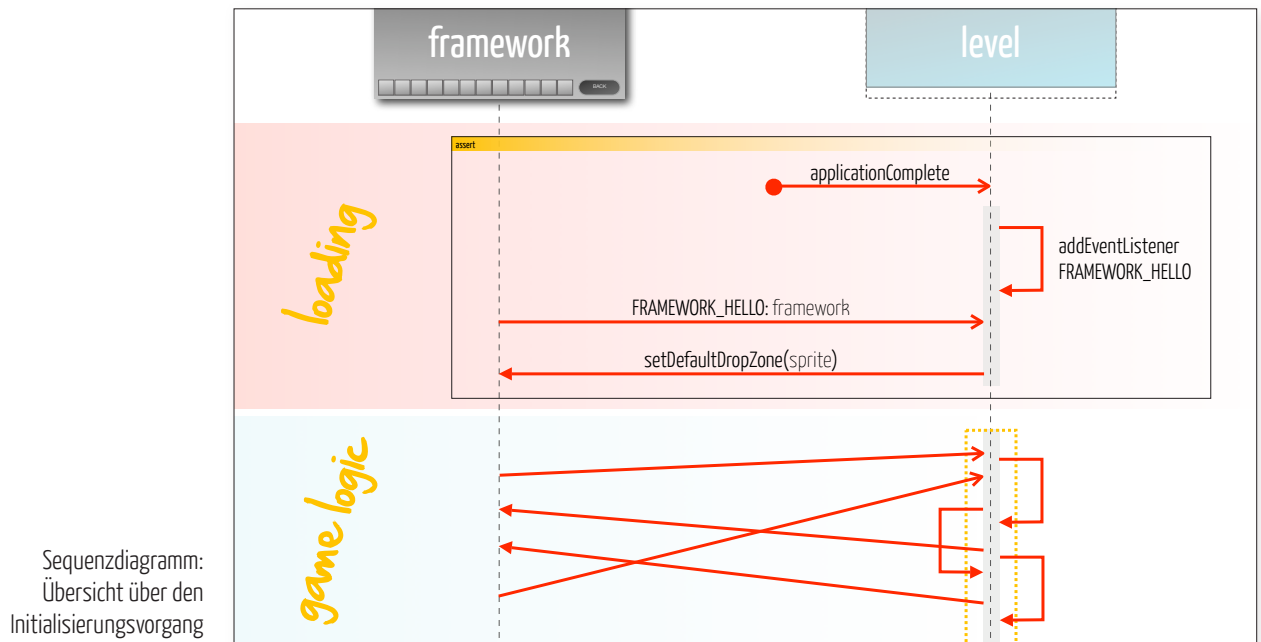
In einem Level werden mehrere Ansichten benötigt. Es ist empfehlenswert, diese Sichten zu kapseln und jeweils die richtige Ansicht in die View-Hierarchie einzugliedern. So ist sichergestellt, dass wirklich nur die Elemente der aktuellen Ansicht zu sehen sind und auch nur damit interagiert werden kann. Ein möglicher Ansatz, der einzelne mxmL-Komponenten für jede Sicht verwendet, ist im Beispiel-Level implementiert. Es kann auch durchaus sinnvoll sein eine Basisklasse für alle Ansichten zu

erzeugen. Wichtig ist, dass dem Framework (für jede Sicht) die Dropzone mitgeteilt wird. Dies ist jenes Sprite, in das das Framework bei Bedarf Items legt.

Die Items an sich können beliebig implementiert werden. Auch hier kann es sich als sinnvoll erweisen, eine eigene Basisklasse dafür zu erstellen. Der Code kann von der eigentlichen Darstellung getrennt sein (MVC-Pattern) oder auch direkt in der grafischen Darstellung integriert sein. Dem Framework muss immer das anzuzeigende Objekt, das von Sprite abgeleitet sein muss, übergeben werden. Daher können Sie fertige Klassen wie `mx.controls.Image` u.a. verwenden, die ihrerseits von `Sprite` abgeleitet sind. Beachten Sie, dass alle Item-Events auf dem darzustellenden Objekt dispatched werden.

Framework Objekt holen

Das Framework fängt den Event `applicationComplete` ihres Levels ab und sendet dann den FrameworkEvent `FRAMEWORK_HELLO`. Dieser FrameworkEvent beinhaltet eine Referenz auf das zu verwendende Framework-Objekt.



Registrieren Sie eine Funktion, die bei `applicationComplete` Ihres Levels ausgeführt wird, um dieses Event abzufangen:

```
<mx:Application ... applicationComplete="appInit();" ...>
```

Fügen Sie in dieser Funktion der Application einen EventListener auf `FRAMEWORK_HELLO` hinzu:

```
private function appInit():void{
    this.addEventListener(
        FrameworkEventType.FRAMEWORK_HELLO, appStart);
}
```

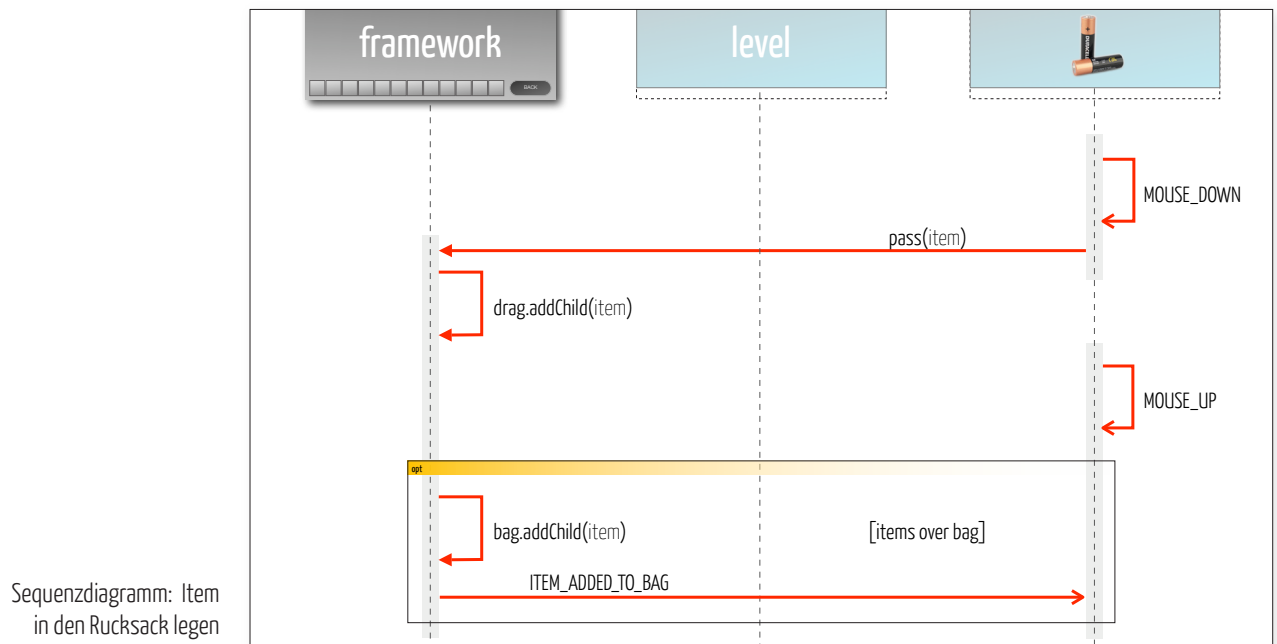
In der Funktion `appStart` erhalten Sie dann das FrameworkEvent und können dort die Referenz auf das Framework abspeichern und erste Einstellungen vornehmen (Debug-Level angeben, etc.):

```
private function appStart(evt:FrameworkEvent):void{
    var fw:Framework = evt.framework;
    // ...
}
```

Bitte beachten Sie, dass Ihr Level erst nach dem Erhalten dieses Events mit dem Ablauf der game-logic beginnen darf. Sie dürfen jedoch bei Bedarf davor schon Vorbereitungen durchführen (Daten laden, Vorberechnungen durchführen, Objekte initialisieren).

Typische Verwendungen

Bei einer typischen Interaktion mit einem Item, wird dieses bei `MOUSE_DOWN` dem Framework mittels der Methode `pass` übergeben. Damit befindet sich das Item über allen anderen Objekten. Wird das Objekt losgelassen, während es sich über dem Rucksack befindet, so wird es in diesen gelegt.

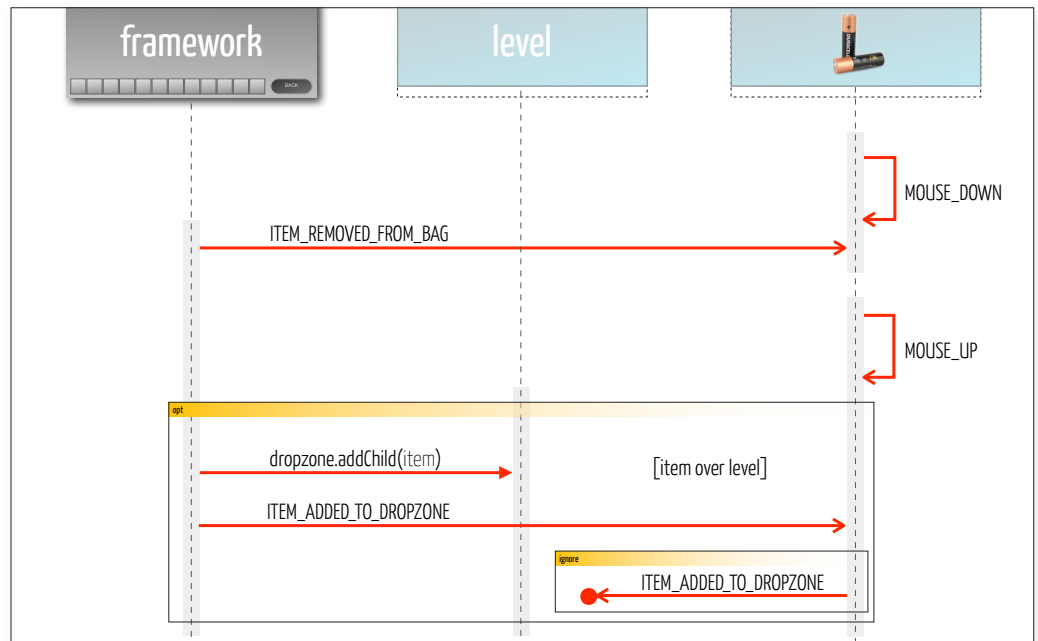


Auf dem Item wird das ItemEvent `ITEM_ADDED_TO_BAG` ausgelöst und die Größe des Objektes an die des Rucksacks angepasst. Bei Erhalt dieses Events kann das Aussehen des Objekts entsprechend angepasst werden, um beispielsweise ein passendes Icon anzuzeigen. Es kann auch erforderlich sein, die Größe von Kinderelementen entsprechend anzupassen, wie beispielsweise im Testlevel bei der Wii-Fernsteuerung. Wird ein Item aus dem Rucksack gezogen, so wird sofort beim Beginn des Drag-Vorgangs auf dem Item das ItemEvent `ITEM_REMOVED_FROM_BAG` ausgelöst und die Größe wieder auf den Originalwert vor dem Hinzufügen in den Rucksack zurückgesetzt. Bei Empfang dieses Events kann bei Bedarf das Aussehen des Items verändert werden, um wieder die im Level erwünschte Darstellung zu erhalten.

Beachten Sie, dass das Item solange es sich über dem Framework befindet, halbtransparent dargestellt wird. Sobald sich der Mauszeiger über dem Level befindet, wird die Transparenz wieder auf den Originalwert zurückgestellt.

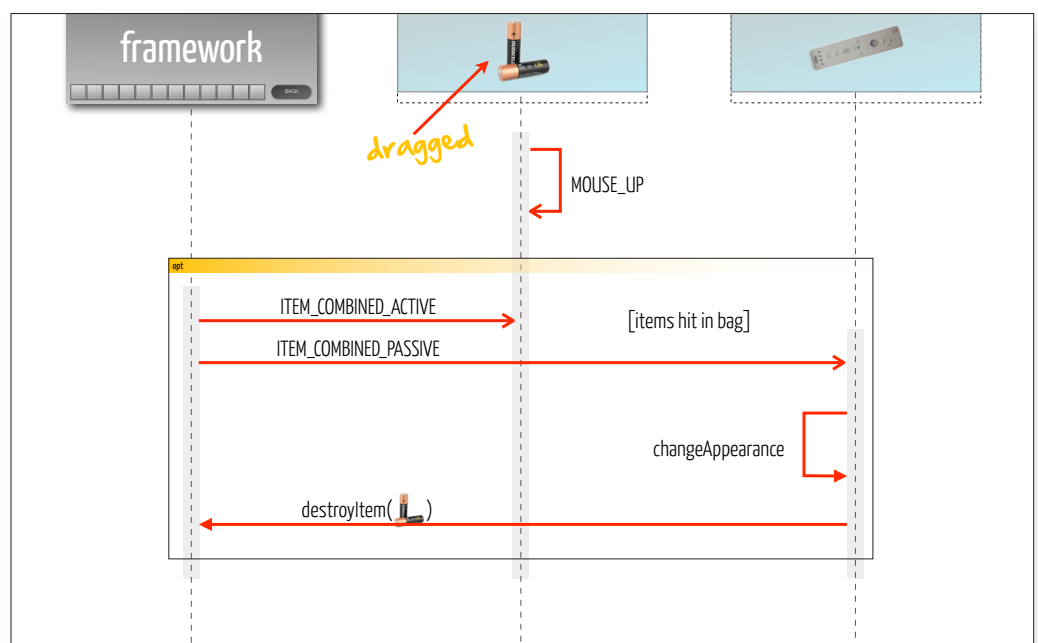
Wird das Item über dem Level losgelassen, so wird es der Dropzone hinzugefügt und ein ItemEvent `ITEM_ADDED_TO_DROPZONE` auf dem Item ausgelöst. Dieses Event bubbled, sprich es steigt in der View-Hierarchie nach oben und kann auch auf der Dropzone abgefangen werden. Als Reaktion auf diesen Event kann das Item zum Beispiel an einer anderen Stelle im Level eingehängt werden, oder auch ein hier nicht erwünschtes Item mittels `insertIntoBag` in den Rucksack zurückgelegt werden.

Sequenzdiagramm: Item aus dem Rucksack nehmen



Das Framework behandelt auch das Kombinieren von zwei Elementen im Rucksack. Wird ein Item auf ein anderes Item im Rucksack gezogen, so erhält das gezogene Item das CombineEvent `ITEM_COMBINED_ACTIVE` und das stationäre Item das CombineEvent `ITEM_COMBINED_PASSIVE`. Das gedraggte Item springt automatisch wieder an seine alte Position zurück. Überprüfen Sie beim Erhalt des CombineEvents, ob diese beiden Elemente kombinierbar sind. Ist dies der Fall, so wird es meist erforderlich sein das Aussehen eines der beiden Gegenstände anzupassen (z.B.: bei der Wii-Fernbedienung nach Einlegen der Batterien den Button leuchten zu lassen). Wird das gedraggte Objekt (z.B. hier die Batterien) nach dem Kombinieren nicht weiter benötigt, so kann es mittels `destroyItem` entfernt werden.

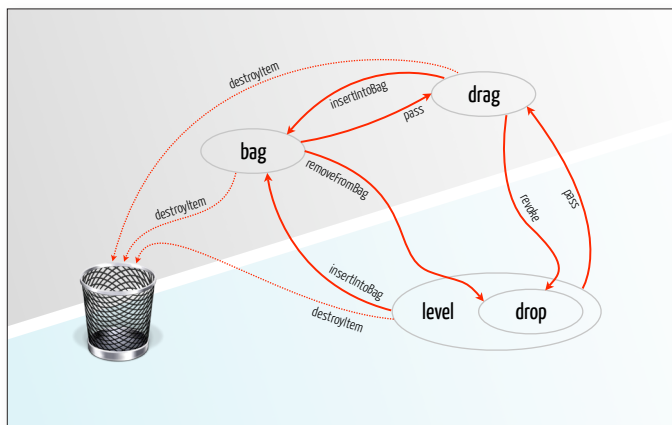
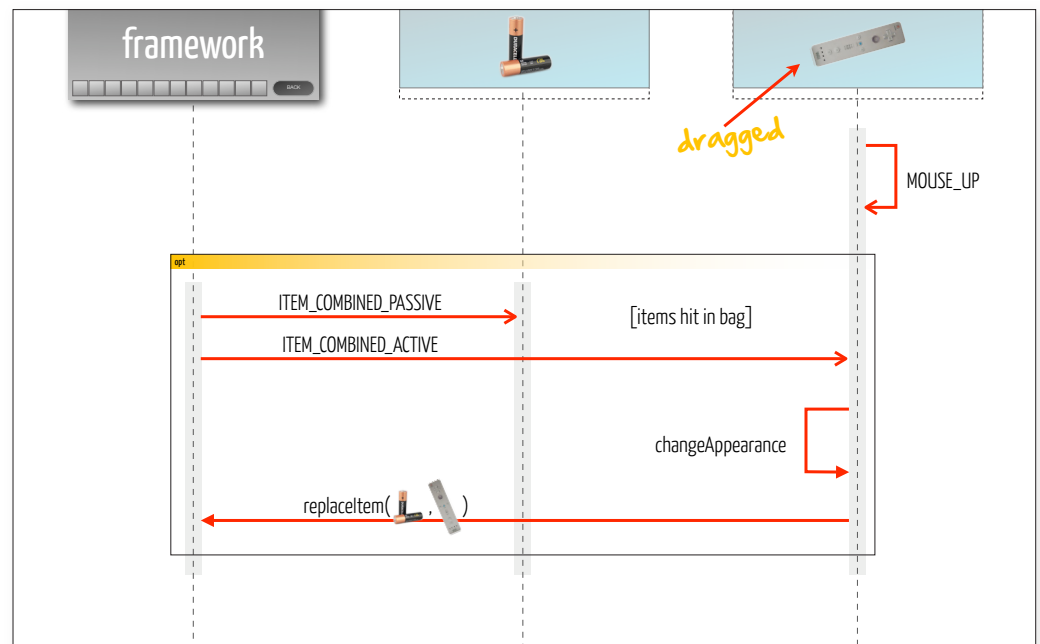
Sequenzdiagramm: Items kombinieren mit anschließendem Entfernen eines der beiden Objekte



Wird das gedraggte Objekt weiterhin benötigt (z.B. die Wii-Fernbedienung), jedoch das stationäre Objekt (die Batterien) nicht, so kann dies

mittels `replace` bewerkstelligt werden. Dabei wird ein Objekt durch ein anderes ersetzt. Dies hat zur Folge, dass das ersetzte Objekt wie bei `destroyItem` verschwindet und das andere Item dessen Position einnimmt.

Sequenzdiagramm:
Items kombinieren mit
anschließendem Ersetzen
eines der beiden Objekte



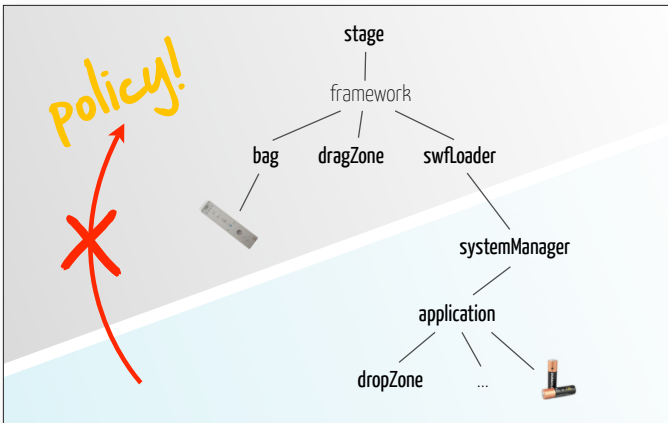
Übersicht über die verschiedenen Zustände, in dem
sich Items befinden können und den Methoden, mit
denen die Zustände gewechselt werden können.

Beschreibung der Events und Funktionen

Ein Item kann drei Zustände einnehmen: Meistens befinden sich Items im Level oder im Rucksack. Maximal ein Item kann gedraggt sein. Zwischen diesen Zuständen kann mit den entsprechenden Methodenaufrufen gewechselt werden, mit den Methoden `isInBag` und `isDraggedInFramework` kann abgefragt werden, ob sich ein Item in einem der beiden Zustände befindet. Wird ein Item an das Level zurückgegeben, so wird es immer der Dropzone hinzugefügt. Items, die sich im Rucksack befinden oder die gerade gedraggt werden und sich über dem Framework befinden (sprich der Mauszeiger ist nicht über dem Level), erhalten keine `MOUSE_UP`, `MOUSE_DOWN` und `MOUSE_MOVE` Ereignisse.

Policy

- » Greifen Sie niemals direkt auf Elemente des Frameworks zu. Sprich verwenden Sie für alle Zugriffe das mittels `FrameworkEvent` übermittelte Objekt. Beachten Sie auch, dass Sie nicht irrtümlich durch Zugriffe in der View-Hierarchie (`parent`, `stage`, etc.) auf das Framework zugreifen oder Veränderungen darin vornehmen.
- » Beachten Sie, dass sich, insbesondere durch Verschieben und Herausnehmen von Items in den und aus dem Rucksack, das Eltern-Element eines Items verändert. Greifen Sie daher nicht auf `item.parent` zu.
- » Entfernen Sie alle Eventlistener, die Sie mit `addEventListener` hinzufügen auch wieder mit `removeEventListener`. Dies gilt auch dann, wenn Ihr Level beendet (fertig gespielt) ist.
- » Beachten Sie, dass ein Item, das an das Framework übergeben wurde, solange es über der Fläche des Frameworks ist, keine der MouseEvents `MOUSE_MOVE`, `MOUSE_UP` oder `MOUSE_DOWN` erhält. So bewegt die Wii-Fernbedienung im Beispiel-Level den Pong-Schläger nur, wenn sich die Fernbedienung über dem Level und nicht über dem Framework



Überblick über die View-Hierarchie: Grau kennzeichnet hierbei das Framework, blau hinterlegt sind die Elemente des Levels. Greifen Sie niemals auf Elemente des Framework direkt zu.

befindet. Analog erhalten Kinderelemente eines Items ebenso keine MouseEvents. Sollten Sie trotzdem diese Ereignisse benötigen, so können Sie entsprechende MouseEvent-Listener mit einer Priorität größer 300 registrieren.

- » Setzen sie unbedingt immer die Dropzone.
- » Items die bereits dem Framework übergeben wurden, müssen, falls sie gelöscht werden sollen, mittels `destroyItem` gelöscht werden, unabhängig davon, ob sie sich momentan im Framework befinden oder nicht. Diese Methode entfernt Objekte aus der View-Hierarchie.
- » Führen Sie keine Interaktionen, Animationen, Wiedergaben aus, bevor Ihr Level das Event `FRAMEWORK_HELLO` erhalten hat.

Updaten des Frameworks

Das Framework überprüft regelmäßig ob eine aktuellere Version des Frameworks existiert und gibt eine entsprechende Information in die Debugkonsole aus. Sie können sich selbst unter <http://igw.tuwien.ac.at/igw/lehre/mp2id/framework/> über den aktuellen Stand informieren. Wenn es eine neuere Version gibt, laden Sie die entsprechende ZIP-Datei herunter und entpacken Sie sie im Verzeichnis `fw` (für Flash-Entwickler im Stammverzeichnis) Ihres Projekts. Stellen Sie dabei sicher, dass die bestehenden Dateien des Frameworks überschrieben werden. Führen Sie danach ein Project-Clean aus. Beim nächsten Ausführen Ihres Programms sollte die entsprechende neue Version in der Debugkonsole aufscheinen. Manchmal kann es erforderlich sein noch andere Veränderungen außerhalb des Ordners `fw` vorzunehmen. Entsprechende Anweisungen entnehmen Sie bitte der Webseite.

Links

Wenn Sie mit Events nicht vertraut sind, lesen Sie die folgenden Artikel:

- » http://livedocs.adobe.com/flex/3/html/help.html?content=events_01.html
- » http://freehandmx.com/de/devnet/flash/quickstart/event_handling_as3/
- » <http://learn.adobe.com/wiki/display/Flex/Event+Listeners>
- » http://edutechwiki.unige.ch/en/ActionScript_3_event_handling_tutorial

Verwenden sie Adobe Flash Player 10, Debugger Version

- » <http://www.adobe.com/support/flashplayer/downloads.html>