Projektarbeit

„Book List“

Andreas Fey, Thomas Flugs, Marco Grohmann
Informatik 3. Semester
fhS

Inhaltverzeichnis

1 Kurzbeschreibung

Bei diesem Projekt handelt es sich um eine Bücherverwaltung. Ein Benutzer wird nach der Begrüßungsseite aufgefordert, den Namen und die Adresse des jeweiligen Bücherbesitzers einzugeben. Die eingegebenen Daten werden sofort in einem Textfeld oberhalb der Buttons angezeigt. Wenn dieser Schritt erledigt ist, kann der Benutzer auf die nächste Seite des Programms wechseln und dort nun beginnen, die Bücher des Besitzers einzugeben. Es sind folgende Angaben eines Buches möglich:

- Titel des Buches
- Autor des Buches
- Verleger des Buches

Nach jeder dieser Eingabe, die einzeln je nach betätigtem Button erfolgt, wird alles in einem wiederum über den Buttons platziertem Textfeld ausgegeben und bei jeder neuen Eingabe aktualisiert. Wenn all dies erledigt ist, oder bei Bedarf auch schon vorher, kann nun der „Save“ – Button betätigt werden, was dazu führt, dass alle Eingaben, so wie sie im obigen Textfeld erscheinen, in die Bücherliste gespeichert werden. Beim Betätigen des „Show“ Buttons wird der jeweils nächste Eintrag angezeigt.

2 Haupteigenschaften

Hier eine Beschreibung der verschiedenen Variablen:

1. Variabeln, die für jedes Buch gleich sind (class variables):
    - `OwnerName;` hier befindet sich der Name des Besitzers dieser Buchliste. Er ist für alle Bücher gleich und kann nur auf der ersten Seite des Programms für alle Bücher global geändert werden.
    - `OwnerAdress;` beinhaltet die Adresse des Besitzers der Buchliste, sie ist ebenfalls für alle Bücher gleich
    - `Index1;` diese Variable ist vom Typ OrderedCollection und beinhaltet die gesamte Buchliste.
2. Variablen, die für jedes Buch verschieden sind (instance variables) und sich so bei jeder Instanz eines Buches unterscheiden:
    - `bookAuthor;` diese Instanzvariable beinhaltet den Autor für das jeweilige Buch und muss jedem Buch einzeln zugewiesen werden.
    - `bookTitle;` genau wie die vorige Instanzvariable ist diese, die den jeweiligen Titel des Buches enthält, für jedes Buch verschieden.
    - `bookPublisher;` diese Instanzvariable enthält den Verleger des jeweiligen Buches.
3. Instanz-Variablen des „Textfield Book“
    - `currentbook;` diese Variable, die nur für die Skripte des Textfield namens „Book“ ihre Gültigkeit hat, enthält die Anzahl der schon gespeicherten Bücher
    - `newbook;` Diese Variable wird als Kontrolle eingesetzt, ob vor dem Speichern auch ein neues Buch erzeugt wurde.

Um die Überschrift auf der ersten beziehungsweise auf der 2. Seite zu erzeugen, wurde ein TextMorph verwendet. Die gesamte Applikation basiert auf einem BookMorph. Auf der zweiten und dritten Seite kommt ein Textfield (auf Seite2: “Owner“, auf Seite3; „Book“) zum Einsatz, welches auf die jeweils mit den verschiedenen Buttons gesteuerten Aktionen reagiert.

3 Ablaufbeschreibung

Der Nutzer startet mit dem Startbildschirm des BookMorph. Dieses Morph ist eine einfach zu handhabende Oberfläche für die Anwendung „Book list“. Die Textfelder geben den Namen des Programms und die seiner Autoren an. Mit diesen Standartbuttons, die auf den oberen Steuerbalken liegen, kann man jederzeit zwischen den einzelnen Anwendungsoberflächen hin - und herschalten.
Nutzt der Endanwender bei dem Startbildschirm den Standartbutton „NextPage“ kommt er auf die Ebene der Nutzerverwaltung. Diese Ebene besteht aus einem Textfeld mit den Daten des aktuellen Nutzers, sowie einigen Schaltflächen mit denen man die Nutzerdaten verändern kann.
Die Schaltflächen „Owner changeName“ und „Change Adress“ öffnen beim Betätigen ein Texteingabefenster. Dieses Fenster zeigt den aktuellen Namen bzw. Adresse des Nutzers an und bietet die Möglichkeit diese Daten zu verändern. Nach Bestätigung schließt das Texteingabefenster und die Veränderungen werden gespeichert.
Eine erneute Betätigung des Standartbutton „NextPage“ auf den oberen Steuerbalken führt den Anwender zu der eigentlichen Buchverwaltung. Diese besteht aus einem Ausgabefenster, welches die Buchdaten wie „Titel“, „Autor“ , „Publisher“ und aktuelle Listennummer darstellt. Darunter sind die Steuerbuttons für die Buchverwaltung angelegt.
So kann man mit den Schaltern „next“ und „previous“ die Liste der angelegten Buch-Datensätze bequem durchschalten und sie sich in den Ausgabefenster anschauen. Es ist auch möglich, neue Bücher hinzuzufügen; dazu kann man den Button „createNewBook“ betätigen. Nach der Betätigung wird ein neuer Buchdatensatz an das Ende der Bücherliste angehangen. Dieser neue Datensatz ist dann bereits mit einem Beispieldatensatz gefüllt, der dem Endanwender als Beispiel dienen kann.
Selbstverständlich kann dieses Beispiel mit den Schaltflächen „Buchtitel“, „Autor“ und „Publisher“ angepasst werden. Die Veränderung kann man in dem Textfenster vornehmen, welches sich bei der Betätigung des entsprechenden Buttons aufbaut. In dem Textfenster wird dann noch mal der bestehende Inhalt des Datensatzes bzw. bei neuen Datensätzen ein Beispiel angezeigt.
Diese Inhalte können dann beliebig überschrieben oder abgeändert werden. Ist der Nutzer zufrieden, speichert er den neuen Datensatz mittels Betätigen des Buttons „save“.
Selbstverständlich hat der Endanwender auch die Möglichkeit einen Buchdatensatz zu löschen. So sucht er mit den Schalter „next“ oder „previous“ den zu löschenden Datensatz im Ausgabefenster aus und betätigt dann den „Remove“-Button. Die Autoren möchten an dieser Stelle darauf hinweisen, dass eine komplett leere Buchliste nicht dem Sinn des Programms entspricht und empfehlen daher mindestens einen Beispieldatensatz zu erhalten, da auch das Programm ein komplettes Löschen der Buchliste nicht unterstützt.

4 Non-visual Part

“Die Klasse Buch:

- Als Instanz-Variablen werden der Buch Autor (bookAuthor), der Titel (bookTitle) und der Verleger (bookPublisher) benutzt.
- Als Klassen-Variablen werden der Name des Besitzers (OwnerName), seine Adresse (OwnerAdress) und eine Variable, um auf die Liste zuzugreifen (Index1) benutzt.“

```
Object subclass: #Book
     instanceVariableNames: 'bookAuthor bookTitle bookPublisher
'
     classVariableNames: 'Index1 OwnerAdress OwnerName '
     poolDictionaries: ''
     category: 'private-Projects'!
```

“Autor des Buchs eingeben”

```
!Book methodsFor: 'settings' stamp: 'af 1/28/2002
10:05'!

setBookAuthor:aString    bookAuthor_aString.
! !
```

“Verleger des Buchs eingeben”

```
!Book methodsFor: 'settings' stamp: 'af 1/28/2002 10:06'!
setBookPublisher:aString
 bookPublisher_aString.
! !
```

“Buch Titel eingeben”

```
!Book methodsFor: 'settings' stamp: 'af 1/28/2002 10:05'!
setBookTitle:aString
 bookTitle_aString.
! !
```

“Initialisiert alle 3 Buch Eigenschaften”

```
!Book methodsFor: 'init' stamp: 'af 1/28/2002 10:04'!
initialise                bookTitle_'Shining'.
  bookAuthor_'Stephen King'.
  bookPublisher_'Heine'.
! !
```

“Speichert das Buch mit allen Eigenschaften”

```
!Book methodsFor: 'init' stamp: 'af 1/29/2002 19:48'!
placeAndSave
     Index1 add: self
! !
```

“Liefert Autor zurück”

```
!Book methodsFor: 'gettings' stamp: 'af 1/28/2002 11:21'!
getBookAuthor
^bookAuthor
! !
```

“Liefert Verleger zurück”

```
!Book methodsFor: 'gettings' stamp: 'af 1/28/2002 11:21'!
getBookPublisher
^bookPublisher
! !
```

“Liefert Titel zurück”

```
!Book methodsFor: 'gettings' stamp: 'af 1/28/2002 11:21'!
getBookTitle
^bookTitle
! !
```

```
"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Book class
     instanceVariableNames: ''!
```

“Liefert Besitzer-Adresse zurück”

```
!Book class methodsFor: 'showing' stamp: 'af 1/28/2002 09:58'!
getAdress
^OwnerAdress.
! !
```

“Liefert gesamte Liste zurück”

```
!Book class methodsFor: 'showing' stamp: 'af 1/29/2002 11:33'!
getIndex
Transcript cr; show:'call getIndex'.
^Index1
! !
```

“Liefert Besitzer-Name zurück”

```
!Book class methodsFor: 'showing' stamp: 'af 1/28/2002 09:58'!
getName        “Liefert Besitzer-Name zurück”
^OwnerName
! !
```

“Setzt den Besitzer zurück auf Standart-Werte”

```
!Book class methodsFor: 'showing' stamp: 'af 1/29/2002 11:33'!
setUp
 OwnerAdress_'FH Schmalkladen'.
 OwnerName_'Harry'.
 Index1_OrderedCollection new.
 ! !
```

“Setzt Besitzer-Adresse”

```
!Book class methodsFor: 'accessing' stamp: 'af 1/28/2002
10:22'!
setAdress: aString
 OwnerAdress_aString.
! !
```

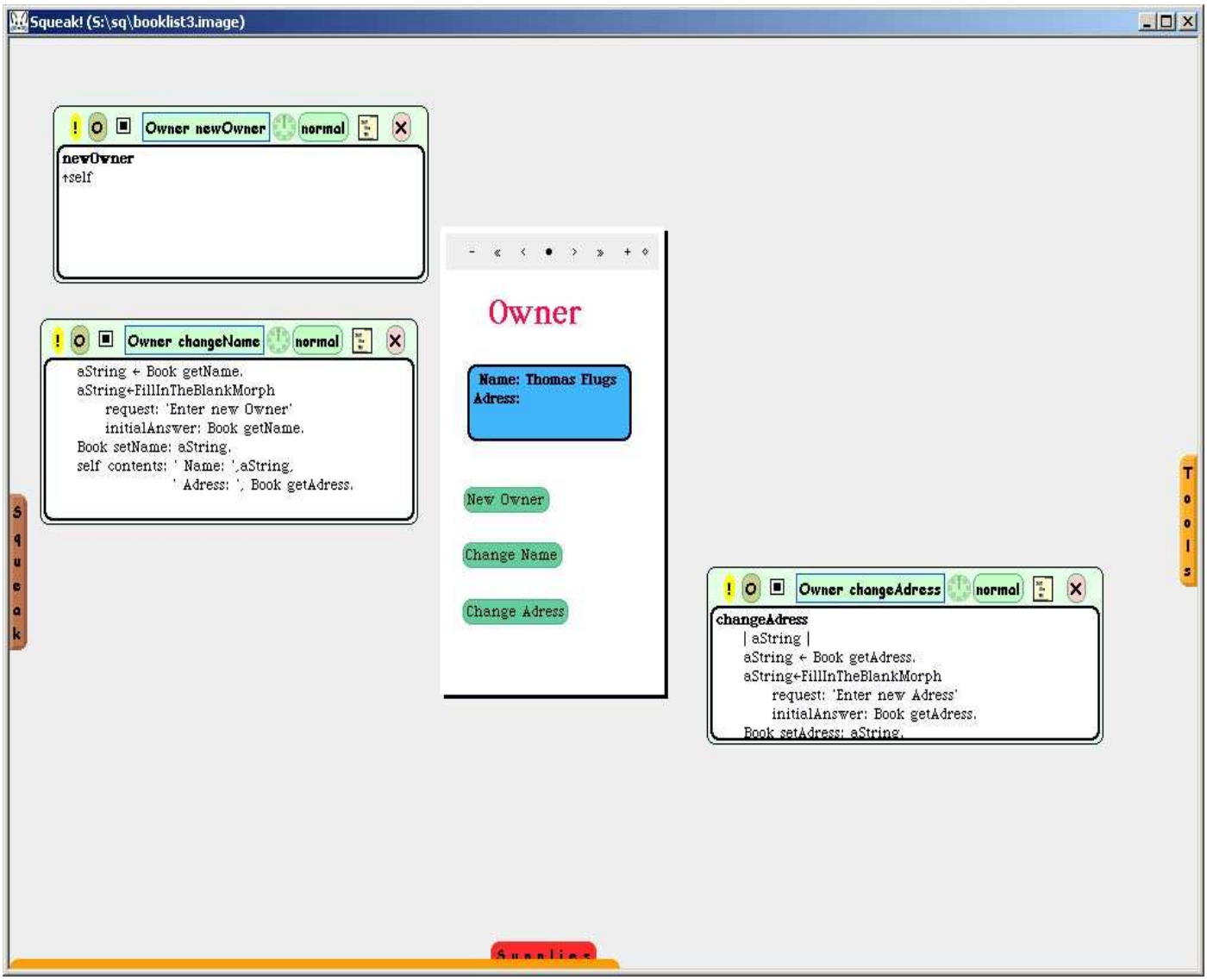
“Setzt Besitzer-Name”

```
!Book class methodsFor: 'accessing' stamp: 'af 1/28/2002
10:22'!
setName:aString
 OwnerName_aString.
! !
```

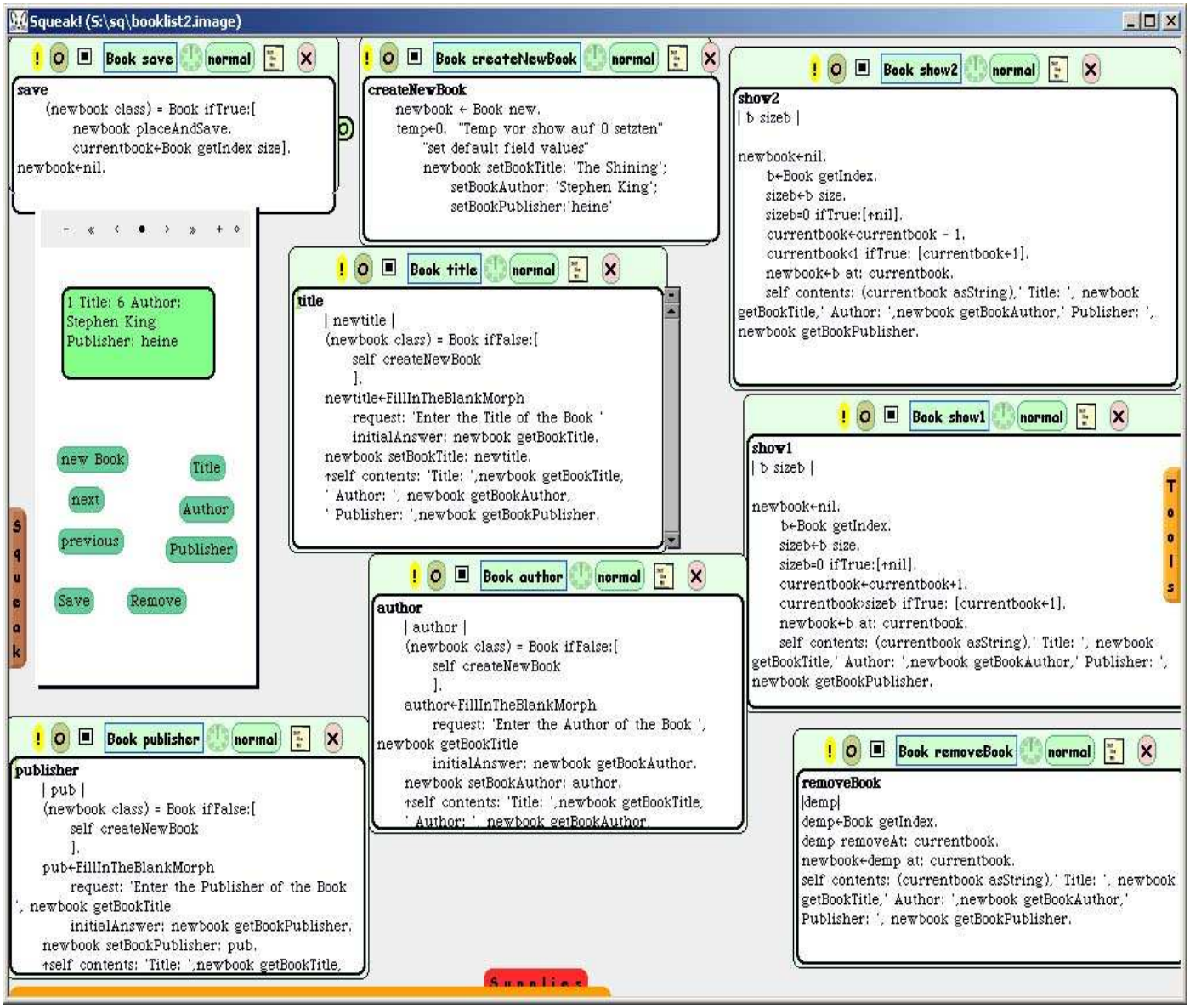
“neuen Besitzer anlagen”

```
!Book class methodsFor: 'initializing' stamp: 'af 1/28/2002
09:56'!
new
^super new initialize.
! !
```

5 Visual Part

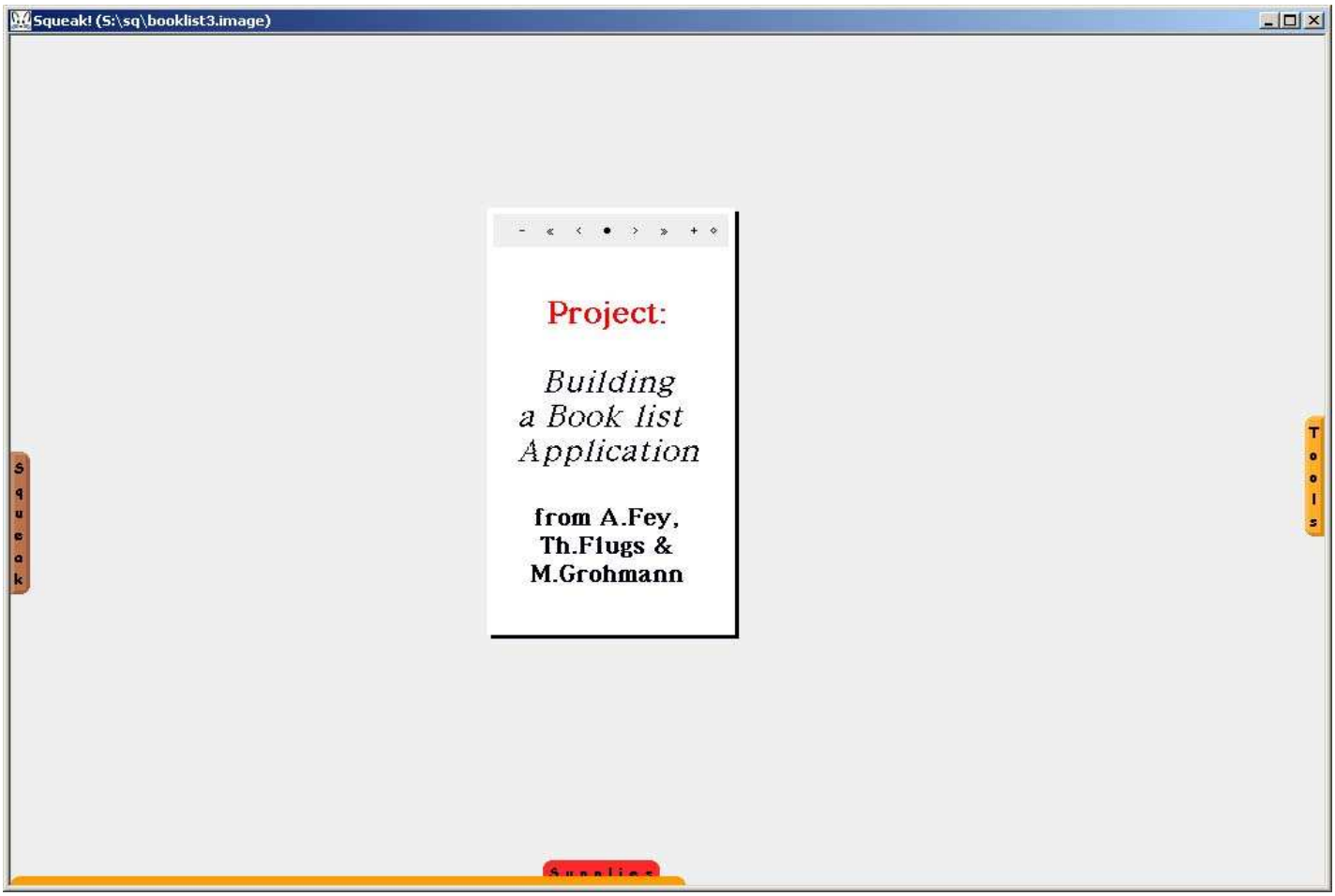


In diesem Screenshot sind die Scripts zu den einzelnen Buttons des BookMorph „Owner“ zu sehen.
Beim drücken des Buttons „Change Name“ wird das Script „Owner changeName“ ausgeführt, welches den Namen eines neuen Nutzers verändert und selbstständig speichert. Dasselbe passiert beim Button „Change Adress“ mit der Adresse des Nutzers; Das Skript „Owner changeAdress“ wird aufgerufen. Ebenso wird beim drücken des Buttons „New Owner“ verfahren.

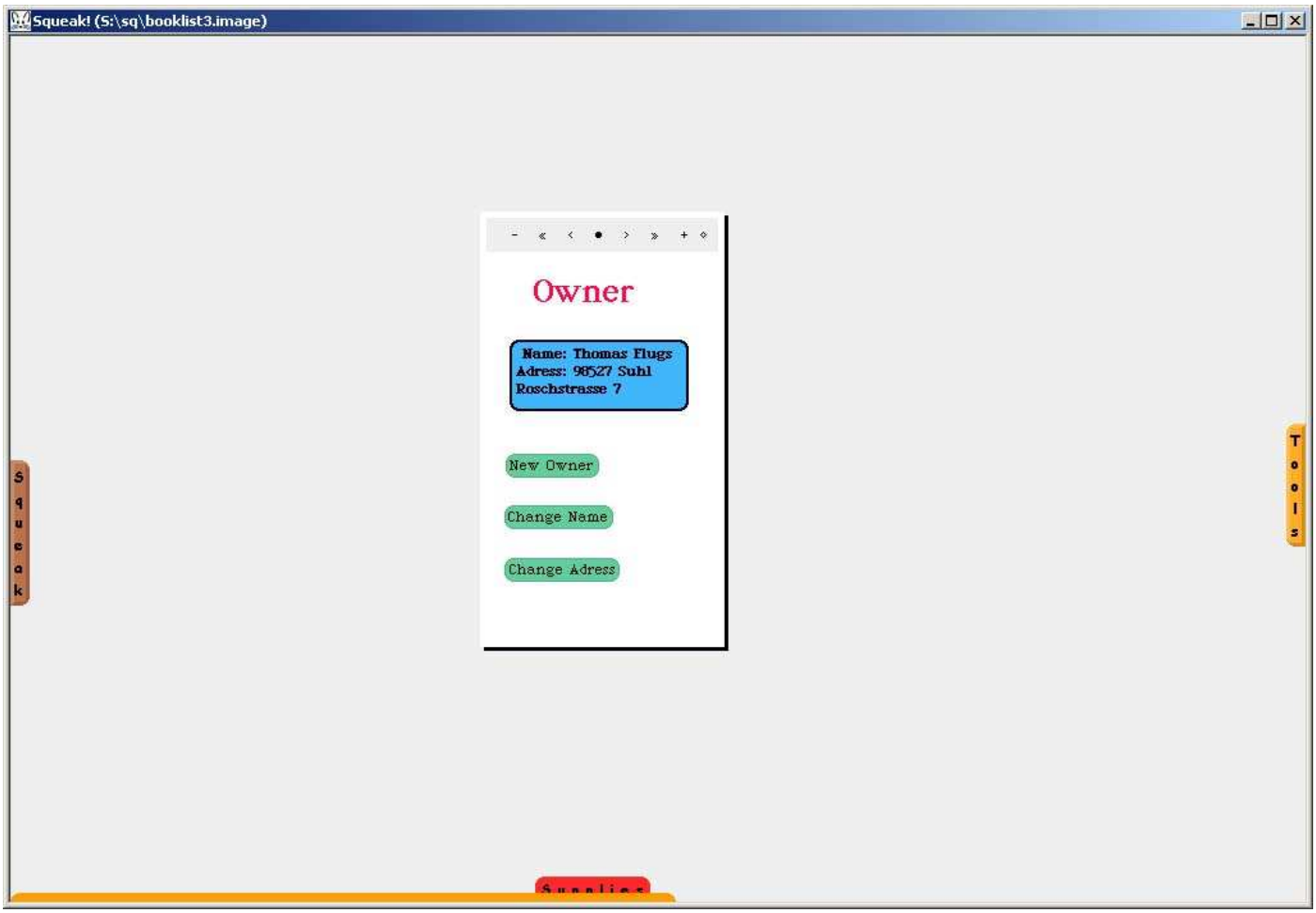


Hier sieht man die Scripts zu den Buttons für die Buchverwaltung. Drückt man den Button „new Book“ wird „createNewBook“ ausgeführt. Dort wird der Buchtitel, der Autor und der Verleger auf eine Voreinstellung gesetzt.
Durch drücken von „Title“ wird das Script „Book title“ ausgeführt, welches entweder einen neuen Namen oder den Voreingestellten Namen setzt. Genauso funktionieren auch die Scripts zu den Buttons „Autor“ und „Publisher“.
Der Button „Save“ speichert die Daten des neuen Buches.
„Remove“ führt den Script „Book removeBook“ aus welches das aktuelle im Textfeld angezeigte Buch aus der Liste löscht.
Drückt man „next“ wird das Script „Book show1“ ausgeführt. Dadurch wird die Buchliste hinaufgeblättert.
Genau das Gegenteil geschieht beim Script „Book show2“, das bei Betätigen von „previous“ ausgeführt wird.

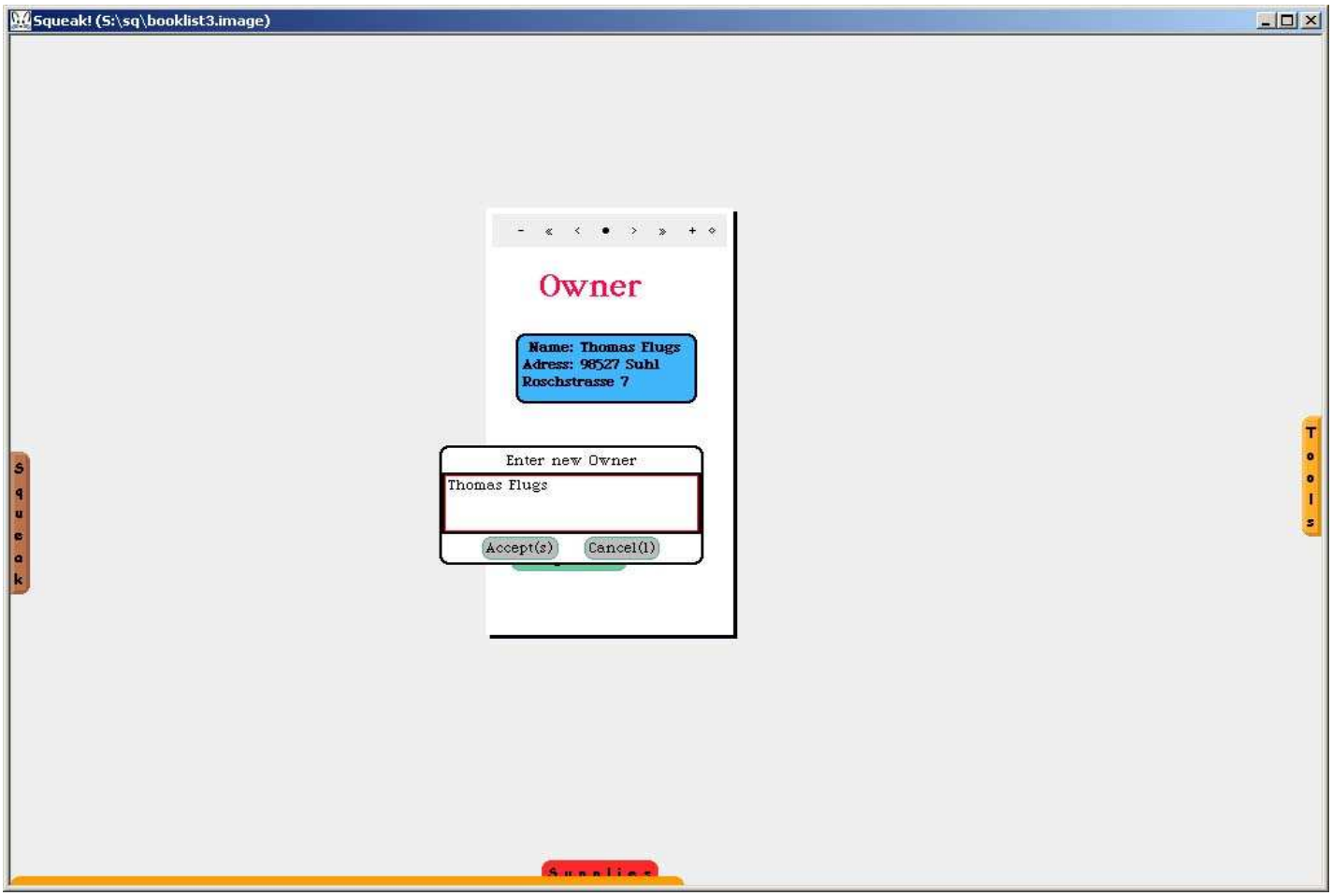
6 Figuren, die unser Projekt illustrieren



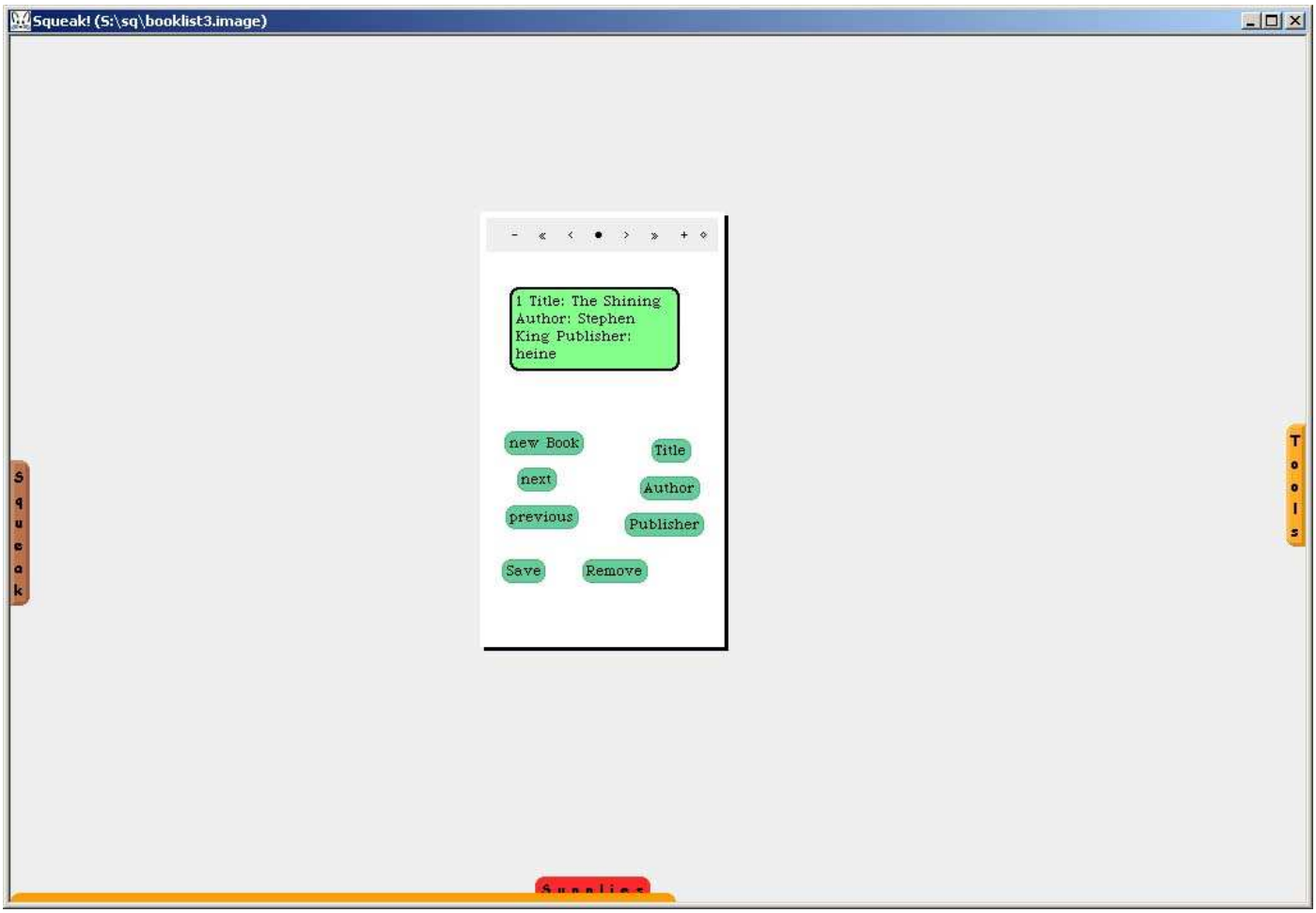
Der Nutzer startet mit diesen Bildschirm. Als Basis wird ein BookMorph verwendet, welches bereits einige Standartbuttons für allgemeine Steuerungsfunktionen in den Oberen Balken enthält.



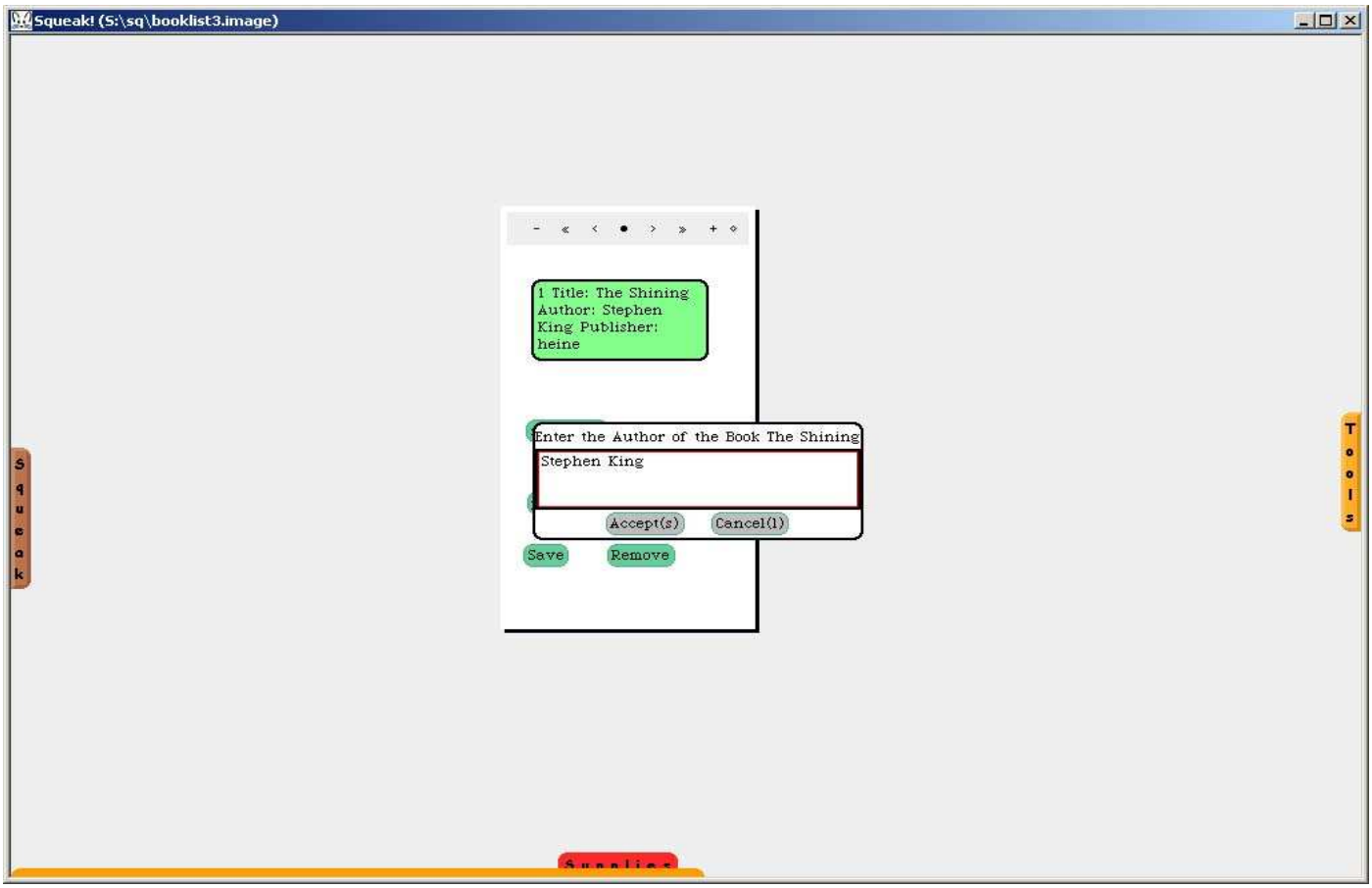
Nutzt der Endanwender bei dem Startbildschirm den Button „NextPage“ in dem oberen Steuerbalken, so kommt er auf diese Ansicht. Hier kann er die Daten des Nutzers der Buchliste verändern, bzw. einen neuen Nutzer festlegen.



Wird einer der Buttons „ „ChangeName“ oder „ChangeAdress“ gedrückt, erscheint Feld bei dem man die neuen Daten einfügen kann, bzw die vorhandenen Datenändern kann. Bereits bestehende Inhalte werden dabei nochmal angezeigt.



Ein weiteres Benutzen des Button „NextPage“ in dem oberen Steuerbalken bringt den Anwender zur eigentlichen Buchverwaltung. Diese Besteht aus einem Ausgabefester bei denen die Daten „Buchtitel“, „Autor“ und „Publisher“ des aktuellen Buches angezeigt werden.
Desweiteren bestehen einige Buttons auf der Anwendung, welche es ermöglichen die Buchliste einfach per Mausklick zu verwalten.



Hier kann man sehen, dass die Verwaltung der Buchliste genauso bequem wie die Verwaltung der Nutzerdaten ist. Nach dem Drücken des Buttons „new Book“ wird ein neuer Buchdatensatz erzeugt welcher sich mit den rechts liegenden Buttos „Titel“, „Autor“ und „Publisher“ editieren lässt. Den ist der Anwender mit dem neuen Datensatz zufrieden, kann er ihn mit dem „Save“-Button speichern.
Des weiteren besteht natürlich die Möglichkeit den aktuellen Buchdatensatz zu löschen, bzw. die Buchliste vor – und zurückzublättern.

7 Quellennachweis

- Vorlesung „Proseminar“ FH Schmalkalden, WS 2001/2002
- Beispielprogramm von H. Markov, „Employee Application“

Kontakt:

andreas@andreas-fey.com
thflugs@gmx.de
grohmann@web.de