```
/****
 *
 * Implementation of rolodex-menu-ui.h  This is a partially stubbed
 * implementation with pulldown menus but no data entry or display dialogs.
 *
 */

#include "edit-menu.h"
#include "file-menu.h"
#include "rolodex-menu-ui.h"
#include "rolodex-menu.h"
#include "rolodex-tool.h"
#include "white-background-tray.h"
#include <InterViews/border.h>
#include <InterViews/glue.h>
#include <InterViews/message.h>
#include <InterViews/tray.h>


RolodexMenuUI::RolodexMenuUI(Screen* s, RolodexTool* rt) :
        RolodexUI(s, rt) {
    /*
     * Construct the file menu, passing its File model.
     */
    fm = new FileMenu(new String("File"), rt->GetFile());

    /*
     * Construct the edit menu, passing its Edit model.
     */
    em = new EditMenu(new String("Edit"), rt->GetEdit());

    /*
     * Construct the edit menu, passing its Edit model.
     */
    rm = new RolodexMenu(new String("Rolodex"), rt->GetRolodex(), this);

    /*
     * For convenience, save a copy of the model as a downcast value.  Hence,
     * this->m == this->rt, where the former is of type Model* and the latter
     * is of type RolodexTool*.
     */
    this->rt = rt;
}

RolodexMenuUI::~RolodexMenuUI() {
}

void RolodexMenuUI::Compose() {

    /*
     * Make the window of this a WhiteBackgroundTray, which isspecialization of
     * an InterViews Tray.  A Tray is a standard outermost interactor for a
     * top-level interface since it provides a opaque background.  An HBox or
     * VBox by itself has a transparent background, which means that if it's
     * used as the outmost interactor, it will "absorb" whatever window
     * background is underneath it.  Hence, it is standard practice to use a
     * Tray for the solid background.  Trays have other useful purposes, but
     * here we're simply using one for its background.
     */
    w = new WhiteBackgroundTray(vb = new VBox());

    /*
     * Compose the menubar and insert it into this' window.  The border
     * provides a visible separator under the menubar.  By default a menubar
     * has no visible border.
     */
    vb->Insert(ComposeMenuBar());
    vb->Insert(new HBorder());

    /*
     * Compose the data entry area and insert into the window.
     */
    vb->Insert(ComposeDataArea());

    /*
     * Make the start-up contents of the data entry empty.
     */
    d->FlipTo(EMPTYPOSITION);
}

MenuBar* RolodexMenuUI::ComposeMenuBar() {
    mb = new MenuBar();

    /*
     * Compose the file menu view and insert its window into the bar.
     */
    mb->Include((PulldownMenu*)ComposeFileMenu()->GetWindow());

    /*
     * Stick some glue between the file and edit menus.
     */
    mb->Insert(new HGlue(
        round(0.2*inch), round(0.15*inch), 0));
     // ^^^^^^^^^^^^^^^^  ^^^^^^^^^^^^^^^^  ^ Stretchibilty,
     // ^^^^^^^^^^^^^^^^  ^^^^^^^^^^^^^^^^
     // ^^^^^^^^^^^^^^^^  ^^^^^^^^^^^^^^^^ Shrinkibility, i.e., minimum size
     // ^^^^^^^^^^^^^^^^
     // ^^^^^^^^^^^^^^^^ Natural size, i.e., initial size

    /*
     * Compose the edit pulldown and insert it into the bar.
     */
    mb->Include((PulldownMenu*)ComposeEditMenu()->GetWindow());

    /*
     * Stick some glue between the edit and rolodex menus.
     */
    mb->Insert(new HGlue(round(0.2*inch), round(0.15*inch), 0));

    /*
     * Compose the rolodex pulldown and insert it into the bar.
     */
```

```
113      mb->Include((PulldownMenu*)ComposeRolodexMenu()->GetWindow());
114
115      /*
116       * Stick some glue on the right edge of the menubar.
117       */
118      mb->Insert(new HGlue(round(0.15*inch), 0, hfil));
119
120      return mb;
121  }
122
123  HBox* RolodexMenuUI::ComposeDataArea() {
124
125      /*
126       * Allocate a deck to hold the operation-specific dialogs.
127       */
128      d = new Deck();
129
130      /*
131       * Allocate and comppose each of the dialogs and insert each into the deck.
132       * Note that the model for each of the dialogs is an appropriate components
133       * of the rolodex tool workspace
134       */
135      ComposeDialogs();
136
137      /*
138       * Put the deck in an HBox with some glue on each end.  This outer hbox
139       * with its glue will allow horizontal stretchibility.  Note here how the
140       * HBox constructor can take more than one argument -- it can take up to 7
141       * arguments.
142       */
143      return new HBox(
144          new HGlue(round(0.2 * inch), 0, hfil),
145          d,
146          new HGlue(round(0.2 * inch), 0, hfil)
147      );
148  }
149
150  FileMenu* RolodexMenuUI::ComposeFileMenu() {
151
152      /*
153       * Compose the file menu, which entails inserting each of its items.
154       */
155      fm->Compose();
156
157      /*
158       * Return the composed menu.
159       */
160      return fm;
161  }
162
163  EditMenu* RolodexMenuUI::ComposeEditMenu() {
164
165      /*
166       * Compose the edit menu, which entails inserting each of its items.
167       */
168      em->Compose();
169
170      /*
171       * Return the composed menu.
172       */
173      return em;
174  }
175
176  RolodexMenu* RolodexMenuUI::ComposeRolodexMenu() {
177
178      /*
179       * Compose the rolodex menu, which entails inserting each of its items.
180       */
181      rm->Compose();
182
183      /*
184       * Return the composed menu.
185       */
186      return rm;
187  }
188
189  void RolodexMenuUI::ComposeDialogs() {
190      /*
191       * Empty.
192       */
193      d->Insert(new Tray());
194  }
195
196  void RolodexMenuUI::ComposeAndInsert(View* v) {
197      v->Compose();
198      d->Insert(v->GetWindow());
199  }
200
201  void RolodexMenuUI::ShowAddCardDialog() {
202      d->FlipTo(ADDPOSITION);
203  }
204
205  void RolodexMenuUI::ShowDeleteCardDialog() {
206      d->FlipTo(DELPOSITION);
207  }
208
209  void RolodexMenuUI::ShowChangeCardDialog() {
210      d->FlipTo(CHANGEPOSITION);
211  }
212
213  void RolodexMenuUI::ShowFindCardDialog() {
214      d->FlipTo(FINDPOSITION);
215  }
216
217  void RolodexMenuUI::ShowFoundDialog() {
218      d->FlipTo(FOUNDPOSITION);
219  }
220
221  void RolodexMenuUI::ShowEmptyDialog() {
222      d->FlipTo(EMPTYPOSITION);
223  }
```