

Improve pini scroll (#34)

Browse files

- Modif gestion scroll lors du refresh du pini
- Quelques corrections et améliorations au passage

- Fix pour le placement de la scrollbar en cas de redessin complet du pini
- Fix bouchot count

master (#34)

dguihal committed on GitHub 19 days ago 1 parent f57b8ca commit 2b859dd516fa8b4076c271db05e4ae53fe6356d1

Showing 4 changed files with 52 additions and 46 deletions.

Unified Split

```

2 core/qbouchot.cpp
@@ -37,7 +37,7 @@ typedef struct QQBouchotDef
37 37
38 38 //Définition des bouchots préconfigurés
39 39 // tiré d'olcc by Chrisix
40 -#define BOUCHOTS_DEF_SIZE 12
41 +#define BOUCHOTS_DEF_SIZE 11
42 QQBouchotDef bouchotsDef[] =
43 {
44     { "dlfp", "http://linuxfr.org/board/index.xml", "http://linuxfr.org/board", "board[message]=sm",

```

```

21 ui/pinipede/qtextbrowser.cpp
@@ -34,6 +34,7 @@ QTextBrowser::QTextBrowser(QString groupName, QPinipede *parent) :
34 34     QTextBrowser(parent),
35 35     m_notifArea(new QNotifArea(this)),
36 36     m_urlHelper(new QPinipedeUrlHelper(this)),
37 +     m_isScrollAtBottom(true),
38 38     m_highlightedNorlogeRef(""),
39 39     m_displayedTotozId(""),
40     m_showUrl(),
@@ -50,6 +51,9 @@ connect(this, SIGNAL(anchorClicked(QUrl)), this, SLOT(onAnchorClicked(QUrl)));
50 51     connect(this, SIGNAL(highlighted(QUrl)), this, SLOT(onAnchorHighlighted(QUrl)));
51 52
52 53
54 +     connect(this->verticalScrollBar(), SIGNAL(valueChanged(int)), this, SLOT(onScrollValueChanged(int)));
55 +     connect(this->verticalScrollBar(), SIGNAL(rangeChanged(int,int)), this, SLOT(onScrollRangeChanged(int,int)));
56 +
57     QTextDocument *doc = document();
58     doc->setUndoRedoEnabled(false);
59     doc->setDocumentMargin(0);
@@ -404,408 +404,408 @@ void QTextBrowser::onExtendedInfoAvailable(QUrl &url, QString &contentType)
404 408     }
405 409     }
406 410
411 +/**
412 + * @brief QTextBrowser::onScrollRangeChanged
413 + * @param min
414 + * @param max
415 + */
416 +void QTextBrowser::onScrollRangeChanged(int min, int max)
417 +{
418 +     Q_UNUSED(min)
419 +     if(m_isScrollAtBottom)
420 +         verticalScrollBar()->setValue(max);
421 +}
422 +
423 +void QTextBrowser::onScrollValueChanged(int value)
424 +{
425 +     m_isScrollAtBottom = (value == verticalScrollBar()->maximum());
426 +}
427 +
428 void QTextBrowser::onThumbnailUrlAvailable(QUrl &url, QString &contentType)
429 {
430     Q_UNUSED(url)

```

```

4 ui/pinipede/qtextbrowser.h
@@ -59,6 +59,8 @@ protected slots:
59 59     void onAnchorClicked(const QUrl &link);
60 60     void onAnchorHighlighted(const QUrl &link);
61 61     void onExtendedInfoAvailable(QUrl &url, QString &contentType);
62 +     void onScrollRangeChanged(int min, int max);
63 +     void onScrollValueChanged(int value);
64 64     void onThumbnailUrlAvailable(QUrl &url, QString &thumbnailUrl);
65 65     void onWebSearchAction();
66 66
@@ -86,6 +88,8 @@ protected slots:
86 88     qreal m_timeUAAreaWidthPx;
87 89
88 90
91 +     bool m_isScrollAtBottom;
92 +
93     QString m_highlightedNorlogeRef;
94     QString m_displayedTotozId;
95     QString m_contextMenuTotozId;

```

```

71 ui/qpinipede.cpp
@@ -199,10 +199,8 @@ void QPinipede::repaintPiniTab(const QString &groupName)
199 199
200 200     QApplication::setOverrideCursor(Qt::BusyCursor);
201 201
202 -     QScrollBar *vScrollBar = textBrowser->verticalScrollBar();
203 -
204 -     bool isAtEnd = (vScrollBar->value() == vScrollBar->maximum());
205 -     int blockNum = textBrowser->cursorForPosition(QPoint(0, 0)).blockNumber();
206 +     int initialBlockNum = textBrowser->cursorForPosition(
207 +         QPoint(0, textBrowser->height())).blockNumber();
208
209     clearPiniTab(groupName);
@@ -232,230 +232,230 @@ void QPinipede::repaintPiniTab(const QString &groupName)
232 230     postWasPrinted = printPostAtCursor(cursor, posts->at(i));
233 231     }
234 232     cursor.endEditBlock();
235 +     textBrowser->repaint();
236 234
237 -     if(isAtEnd)
238 -         vScrollBar->setSliderPosition(vScrollBar->maximum());
239 -     else
240 +         // Redeplacement de la scrollbar pour se retrouver la ou on etait avant
241 +         QScrollBar *vScrollBar = textBrowser->verticalScrollBar();
242 +         vScrollBar->triggerAction(QAbstractSlider::SliderToMaximum);
243 +         int currBlockNum = textBrowser->cursorForPosition(
244 +             QPoint(0, textBrowser->height())).blockNumber();
245 +         while(currBlockNum > (initialBlockNum + 1))
246 +         {
247 -             int currBlockNum = textBrowser->cursorForPosition(QPoint(0, 0)).blockNumber();
248 -             while(currBlockNum < blockNum)
249 -             {
250 -                 vScrollBar->triggerAction(QAbstractSlider::SliderSingleStepAdd);
251 -                 currBlockNum = textBrowser->cursorForPosition(QPoint(0, 0)).blockNumber();
252 -             }
253 -             while(currBlockNum > blockNum)
254 -             {
255 -                 vScrollBar->triggerAction(QAbstractSlider::SliderSingleStepSub);
256 -                 currBlockNum = textBrowser->cursorForPosition(QPoint(0, 0)).blockNumber();
257 -             }
258 +             vScrollBar->triggerAction(QAbstractSlider::SliderSingleStepSub);
259 +             currBlockNum = textBrowser->cursorForPosition(QPoint(0, textBrowser->height())).blockNumber();
260 +         }
261
262     QApplication::restoreOverrideCursor();
@@ -959,49 +952,41 @@ unsigned int QPinipede::insertPostToList(QListPostPtr *listPosts, QPost *post
959 952     ///
960 953     void QPinipede::newPostsAvailable(QString groupName)
961 954     {
962 -         //qDebug() << Q_FUNC_INFO << "from : " << groupName;
963 -
964 -         //On est obligé de locker pour éviter la pagaille dans le pini.
965 -         // On est obligé de locker plus fin pourrait être obtenu en implémentant un lock par groupe
966 +         // On est obligé de locker pour éviter la pagaille dans le pini.
967 +         // On est obligé de locker plus fin pourrait être obtenu en implémentant un lock par groupe
968 +         while(! m_newPostsAvailableMutex.tryLock(1000))
969 +             qWarning() << Q_FUNC_INFO << groupName << "tryLock timeout";
970
971     QTextBrowser *textBrowser = m_textBrowserHash.value(groupName);
972     QScrollBar *vScrollBar = textBrowser->verticalScrollBar();
973
974     QListPostPtr newPosts;
975     foreach(QBouchot *b, QBouchot::listBouchotsGroup(groupName))
976     {
977         QListPostPtr newPostsBouchot = b->takeNewPosts();
978         if(newPostsBouchot.size() > 0)
979     {
980         //qDebug() << Q_FUNC_INFO << "newPosts from : " << b->name();
981         newPosts.append(newPostsBouchot);
982     }
983
984 -         //Au cas ou on serait deja passe avant (cas du signal multiple)
985 +         //Au cas ou on serait deja passe avant (cas du signal multiple)
986 +         if(newPosts.size() == 0)
987 +         {
988 +             m_newPostsAvailableMutex.unlock();
989 +             return;
990 +         }
991
992 -         //Prise en compte des changements de parametre
993 -         //On ne peut pas vérifier les valeurs dans le "printPostAtCursor", trop couteux, du coup on met a jour ici
994 +         //Prise en compte des changements de parametre
995 +         //On ne peut pas vérifier les valeurs dans le "printPostAtCursor", trop couteux, du coup on met a jour ici
996 +         QTextDocument *doc = textBrowser->document();
997 +         updatePiniDisplaySettings(doc);
998
999 -         //Il ne sert a rien d'insérer plus que de posts que le max de l'historique
1000 -         //Il ne sert a rien d'insérer plus que de posts que le max de l'historique
1001 +         while(newPosts.size() > m_maxHistorySize)
1002 +             newPosts.removeFirst();
1003
1004 -         bool wasAtEnd = (vScrollBar->value() == vScrollBar->maximum());
1005
1006     // Tri necessaire puisqu'on a potentiellement melange les posts de plusieurs tribunes
1007     qSort(newPosts.begin(), newPosts.end(), postComp);
1008
1009 -         //On signale via la forme de la souris qu'un traitement est en cours
1010 +         //On signale via la forme de la souris qu'un traitement est en cours
1011     QApplication::setOverrideCursor(Qt::BusyCursor);
1012
1013     QTextCursor cursor(doc);
@@ -1021,7 +1006,6 @@ void QPinipede::newPostsAvailable(QString groupName)
1021 1006     postWasPrinted = printPostAtCursor(cursor, firstPost);
1022 1007
1023 1008     // Si le pini était vide on scrolle forcément vers le dernier post
1024 -     wasAtEnd = true;
1025
1026     }
1027     else
1028     destlistPosts = m_listPostsTabMap[groupName];
@@ -1037,10 +1021,10 @@ void QPinipede::newPostsAvailable(QString groupName)
1037 1021     if(newPost == destlistPosts->last()) //insertion a la fin
1038 1022         break;
1039 1023
1040 -         //Déplacement vers la nouvelle ligne
1041 +         //Déplacement vers la nouvelle ligne
1042 +         if(insertIndex == 0)
1043 +         {
1044 -             //Necessite de le copier car il sera supprime par le nouveau userData de la premiere ligne
1045 -             //Necessite de le copier car il sera supprime par le nouveau userData de la premiere ligne
1046 +             QQMessageBlockUserData * uData = new QQMessageBlockUserData* ((QQMessageBlockUserData *) cursor.
1047 +                 if(postWasPrinted)
1048 +                 cursor.insertBlock();
1049
1050     baseInsertIndex = insertIndex;
1051     }
1052
1053     //Insertion a la fin
1054     //Insertion a la fin
1055     if(newPostsIndex < newPosts.size())
1056     {
1057 -         //Le premier item a deja ete insere dans la liste destlistPosts dans la boucle while au dessus
1058 -         //On a juste a l'afficher
1059 +         //Le premier item a deja ete insere dans la liste destlistPosts dans la boucle while au dessus
1060 +         //On a juste a l'afficher
1061 +         cursor.movePosition(QTextCursor::End);
1062 +         if(postWasPrinted)
1063 +             cursor.insertBlock();
1064
1065     }
@@ -1088,7 +1072,7 @@ void QPinipede::newPostsAvailable(QString groupName)
1088 1072     }
1089 1073     }
1090 1074
1091 -     if(!postWasPrinted) //remove last inserted block if not printed
1092 +     if(!postWasPrinted) //Supprime du dernier bloc insere si non insere
1093     {
1094         cursor.movePosition(QTextCursor::PreviousBlock, QTextCursor::KeepAnchor);
1095         cursor.movePosition(QTextCursor::EndOfBlock, QTextCursor::KeepAnchor);
@@ -1100,10 +1084,10 @@ void QPinipede::newPostsAvailable(QString groupName)
1100 1084     // Purge des anciens messages
1101 1085     purgePiniTabHistory(groupName);
1102 1086
1103 -     if(wasAtEnd)
1104 -         vScrollBar->setSliderPosition(vScrollBar->maximum());
1105
1106 -         //Remise en place de l'ancienne forme du pointeur
1107 +         //Remise en place de l'ancienne forme du pointeur
1108 +         QApplication::restoreOverrideCursor();
1109
1110 -         //Signalement de nouveaux posts dans le nom du Tab
1111 +         //Signalement de nouveaux posts dans le nom du Tab
1112 +         int i = 0;
1113 +         foreach(QPost *p, newPosts)
1114 +         {

```

0 comments on commit 2b859dd

Please sign in to comment.